Instituto Politécnico Nacional

Number 43 January - June 2011

Número 43 Enero - Junio 2011

polibits

Since 1989

THEMATIC ISSUE: COMPUTATIONAL LINGUISTICS AND INTELLIGENT TEXT PROCESSING

Guest Editor: Yasunari Harada

Journal on Research and Development in Computer Science and Engineering
Published by *Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional,* Mexico

Revista de Investigación y Desarrollo Tecnológico en Computación
Publicado por el Centro de Innovación y Desarrollo Tecnológico en Cómputo, del Instituto Politécnico Nacional, México

ISSN: 1870-9044

"La Técnica al Servicio de la Patria"

Printing
Impresión

Estevan Becerril Camargo
Calle Bolivar 118, Col. Centro, C.P. 06080
Delegación Cuahutémoc, México D.F.

Distribution
Distribución

CIDETEC
Av. Juan de Dios Bátiz s/n, casi esq. Miguel Othón de Mendizábal. Unidad Profesional "Adolfo López Mateos", Colonia Nueva Industrial Vallejo, Deleg. Gustavo A. Madero, C.P. 07700, México, D.F.
Teléfono 5729-6000 ext.52513.

Indexing: Latindex, Periódica, e-revistas, Index of Excellence of CONACYT (Mexico)

Polibits. Es una publicación semestral, del Instituto Politécnico Nacional, editada por el Centro de Innovación y Desarrollo Tecnológico en Cómputo (CIDETEC), de la Secretaría de Investigación y Posgrado. El origen y contenido de los artículos publicados en esta revista es responsabilidad exclusiva de los autores y no representa necesariamente el punto de vista del CIDETEC o del Instituto, a menos que se especifique lo contrario. Se autoriza la reproducción parcial o total, siempre y cuando se cite explícitamente la fuente. Domicilio de la publicación: Av. Luis Enrique Erro s/n, Unidad Profesional "Adolfo López Mateos", Zacatenco, Deleg. Gustavo A. Madero, C.P. 07700, D.F., México. Teléfono 5729-6000 ext.52513. Número de Certificado de Reserva otorgado por el Instituto Nacional del Derecho de Autor: 04-2006-063010513100-102. Número de ISSN: 1870-9044, Número de Certificado de Licitud de Título: 13710, Número de Certificado de Licitud de Contenido: 11283. El presente número fue editado en el CIDETEC, en junio de 2011, y tuvo un tiraje de 500 ejemplares.

Editorial

It is my great pleasure to congratulate the readers of the journal and the members of its Editorial Board with a long-awaited success—the inclusion of the journal in the Index of Mexican research journals maintained by the National Council of Science and Technology (CONACYT) of Mexico—the country where this journal is published—in "recognition of the quality and editorial excellence" of the journal, according to the CONACYT. This opens a new page in the 22-year-long history of the journal.

This thematic issue is devoted to computational linguistics and intelligent text processing, a rapidly growing and dynamic field that lays at the intersection of linguistics, artificial intelligence, and computer science. It embraces a variety of technological developments that enable computers to meaningfully process human language—the language that we use both for our everyday communication and to record all human knowledge—in its written form as far as text processing is concerned. Its main applications include search and information retrieval, machine translation, and human-computer interaction, among others.

The first four papers included in this thematic issue deal with semantics of natural language.

The paper "*Detecting derivatives using specific and Invariant descriptors*" by F. Poulard *et al*. from France suggests a faster and simpler method to detect semantic similarity between textual documents that can indicate plagiarism or copying. This task is very important in many application areas, from education to law and forensics. The authors have built a French corpus based on Wikinews revisions in order to facilitate the evaluation of plagiarism detection algorithms for French. Both the corpus and the implementation of their algorithm has been made freely available to the community—an excellent example most research papers that seeks verifiability and the reproducibility of its results should certainly follow.

The paper "*Assessing the feature-driven nature of similarity-based sorting of verbs*" by P. Öztürk *et al*. from Norway presents a computational analysis of the results from a sorting task with motion verbs in Norwegian, which is a contribution to both computational linguistics and psycholinguistics. The authors argue for that when sorting words, humans first compare the words by their similarity, which, in turn, involves comparison of some features of words. The authors investigate the set of these features and show that some of these features are more important than others for human judgments. What is more, they model these features computationally by finding a set of features that give automatic clustering similar to the clustering made by human annotators.

The paper "*Semantic textual entailment recognition using UNL*" by P. Pakray *et al*. from India and Japan describes a system for recognizing textual entailment, i.e., a semantic relation between two phrases consisting in that one of them logically imply the other, e.g.: *John's assassinator was caught by police* ⇒ *John is dead*. This task is crucial in information retrieval, machine translation, text understanding, text summarization and many other tasks and applications of natural language processing. To compare the semantics of the two phrases, the authors use a particular semantic representation originally introduced for machine translation and having its roots in the Meaning ⇔ Text theory: Universal Networking Language (UNL).

The next paper, still continuing the topic of semantics, starts the series of four papers that deal with multilingualism and machine translation. The last paper of the issue can also be included in this group.

The paper "*Examining the validity of cross-lingual word sense disambiguation*" by E. Lefever and V. Hoste from Belgium is devoted to word sense disambiguation, which is the task of automatically determining the intended meaning of a word from the context, e.g.: *a saving account in the bank* vs. *a low wooden bank* vs. *a high bank of the river*. The authors introduce a multilingual approach to the word sense disambiguation task. Instead of using a predefined monolingual sense-inventory such as WordNet, the authors' language-independent framework includes a manually constructed gold standard corpus with word senses made up by their translations in other languages. Experiments with five European languages are reported.

The paper "*Knowledge expansion of a statistical machine translation system using morphological resources*" by M. Turchi and M. Ehrmann *et al*. from Italy shows how to efficiently expand the existing knowledge of a phrase-based statistical machine translation system with limited training parallel data by using external morphological resources. This is a move to a long-awaited combination of statistical and knowledge-based techniques in computational linguistics and machine translation. The authors suggest that their knowledge expansion framework is generic and could be used to add other types of information to the model.

The paper "*Low cost construction of a multilingual lexicon from bilingual lists*" by L. T. Lim *et al*. from Malaysia suggests a low cost method for constructing a multilingual lexicon using only simple lists of bilingual translation mappings. Their method is especially suitable for under-resourced language pairs—which is still the majority of world's languages—because such bilingual resources are often freely available and easily obtainable from the Internet or

conventional paper-based dictionaries. With very little effort the suggested method generates multilingual lexicons of usable quality.

The paper "*A cross-lingual pattern retrieval framework*" by M.-H. Chen *et al*. from Taiwan R.O.C. is aimed at helping people to learn foreign languages by automatically extracting from a text corpus predominant usage patterns that resemble phrases in grammar books, as well as abstract-to-concrete hierarchy of words resembling a thesaurus. These structures effectively assist the process of language learning, especially in manual sentence translation or composition, accelerating lexicographers' and language learners' navigation through and grasp upon the word usages in other languages.

Along with the previous paper, the next two papers discuss morphological and syntactic aspects of text analysis. Specifically, the next two papers focus on less-then-well-studied Indian languages.

The paper "*Clause boundary identification using classifier and clause markers in Urdu language*" by D. Parveen *et al*. from India presents the identification of clause boundary for the Urdu language. This language is spoken by 60 million people and is mutually intelligible with Hindi, which is understood by 400 million people, which makes it the fourth largest language in the world. This language is an official language in Pakistan and five states of India. The authors use statistical classification methods such as conditional random fields.

The paper "*External sandhi and its relevance to syntactic treebanking*" by S. Kolachina *et al*. from India addresses a very interesting linguistic phenomenon characteristic of many Indian languages but found in some other languages, too: major sound changes that occur at word boundaries, so that two words together sound quite differently than each one would sound independently. A phenomenon is English slightly resembling the one addressed in this paper can be exemplified by the change "a" + "action" → "an action", with the difference that with languages with external sandhi such changes are very complicated and ubiquitous, causing the occurrence of forms which are not morphologically analyzable, which poses a problem for all kinds of natural language processing applications. The paper discusses the implications that this phenomenon has for the syntactic annotation of sentences in Telugu, a language with agglutinative morphology, one of official languages of India spoken by 130 million people.

The next three papers are devoted to search, information retrieval, and Internet.

The paper "*Keyword identification within Greek URLs*" by M.-A. Vonitsanou *et al*. from Greece addresses the part of the problem of web information retrieval related to analysis of keywords contained within Internet addresses (URLs). For example, the URL "http://www.greeceturism.com" suggests that the site might be relevant to queries about Greece and about tourism, if these keywords are correctly identified by the search engine. In this specific paper the authors address the problem of identifying such keywords from a language that does not use the Latin alphabet, namely, the Greek language, so people have to employ different transliteration heuristics and habits to encode words in URLs.

The paper "*Contextual analysis of mathematical expressions for advanced mathematical search*" by K. Yokoi *et al*. from Japan is a small step towards a great goal: the possibility to automatically manipulate mathematical knowledge currently existing in the form of free-text mathematical writing in mathematical papers or books. Mathematics is perhaps the most formal and well-defined of all sciences and is apparently very suitable for automatic reasoning. Ironically, mathematical texts are currently nearly unintelligible to computer programs, to such degree that finding a simple mathematical fact in a collection of specialized papers the same way as we find simple everyday-life-related facts in Internet is nearly impossible. The paper addresses this highly important issue and shows how mathematical language can be analyzed and used to resolve search queries.

The paper "*Semantic aspect retrieval for encyclopedia*" by C. Han *et al*. from China describes a method of retrieving passages that are semantically related to a short query, usually consisting of one word or phrase, from a given article in an online encyclopedia. The method takes into account the surrounding snippets of the keywords in the encyclopedia article, which the authors show to give better results than traditional methods that do not take into account the context.

Finally, three of the following four papers present more practical-related applications: style analysis, information extraction, and text summarization. The last two papers, in addition, present artificial intelligence algorithms useful in natural language processing.

The paper "*Are my children old enough to read these books? Age suitability analysis*" by F. Wanner *et al*. from Germany presents a system that analyzes the text of a book and in order to recommend it or not, for reading by children of specific age. The paper's emphasis is not on the traditional issues of parental control, such as sex- or violence-related contents; instead, the paper deals with understandability of the text for children of specific age. As features, it uses linguistic complexity, story complexity, genre, and the like. To compensate for the limitations of automatic methods, it presents a tool that visualizes these features and gives the user the possibility to explore the analysis results.

The paper "*Linguistically motivated negation processing: an application for the detection of risk indicators in unstructured discharge summaries*" by C. Hagege from France, addresses a practical application of information extraction: automatic detection of cases of hospital acquired infections by processing unstructured medical discharge summaries. A particular challenge in this task is highly accurate handling of negation in order to understand whether a potential risk indicator is attested positively or negatively in the text. The author proposes a linguistically motivated

approach for dealing with negation using both syntactic and semantic information.

The paper "*A micro artificial immune system*" by J. C. Herrera-Lozada *et al*. from Mexico considers the task of numerical optimization. The problem of optimization often appears in natural language processing tasks, such as word sense disambiguation or machine translation. The authors propose a fast version of a well-known artificial immune system algorithm, CLONALG. While the standard version of this algorithm tend to suffer from a huge growth of its population size, the authors show that a very small population—in other words, very few calculations of the function being optimized—is sufficient for obtaining good results.

Finally, the paper "*A graph-based approach to cross-language multi-document summarization*" by F. Boudin *et al*. from France, Canada, and Mexico is also based on an artificial intelligence technique. It proposes an improved method for cross-language summarization, i.e., the task of generating a summary in a language different from the language of the source documents, in the settings in which many different documents are reduced to one summary. For this, they integrate machine translation quality scores in the sentence extraction process.

The papers selected for publication in this thematic issue will give the reader a wide panorama of the methods currently used in computational linguistics and intelligent text processing.

Yasunari Harada
Professor,
Director of the Institute for Digital
Enhancement of Cognitive Development,
Waseda University, Tokyo, Japan;
President of the Logico-Linguistics Society of Japan

Detecting Derivatives using Specific and Invariant Descriptors

Fabien Poulard, Nicolas Hernandez, and Béatrice Daille

***Abstract*—This paper explores the detection of derivation links between texts (otherwise called plagiarism, near-duplication, revision, etc.) at the document level. We evaluate the use of textual elements implementing the ideas of specificity and invariance as well as their combination to characterize derivatives. We built a French press corpus based on Wikinews revisions to run this evaluation. We obtain performances similar to the state of the art method (n-grams overlap) while reducing the signature size and so, the processing costs. In order to ensure the verifiability and the reproducibility of our results we make our code as well as our corpus available to the community.**

***Index Terms*—Textual derivatives, detection of derivations, near-duplicates, revisions, linguistic descriptors, French corpus.**

I. Introduction

BEING in the age of information, the information is not only produced but also duplicated, revised and plagiarized at some extent. This redundancy is an hindrance to Information Retrieval (IR) methods in terms of computation, storage and results. Hence, the performance of web search engines could be improved with the filtering of duplicate texts as, meanwhile saving the storage necessary for the index. Moreover, users may not want duplicated (or even near-duplicated) documents in the answer to their search query.

We address the task of detecting text derivatives of a given source document among a collection of suspicious documents, *i.e.* given a collection of suspicious and source documents, one must map the first to the second therefore detecting the derivation links involving a suspicious and a source. This task is usually handled by measuring the n-grams overlap between sources and suspicious. We propose to use textual elements implementing the ideas of specificity and invariance (hapax n-grams, named entities and nominal compounds) instead of n-grams. We report the performance of the classic approach on a corpus we made out of revisions of French news articles. We compare the performances of our propositions to this baseline.

First we introduce the classic signature approach to the problem (Section II). Then we describe the way we built a French corpus (Section III) and present our methods (Section IV) and the evaluation protocol for our experiments (Section V). Lastly, we report the results of our experiments (Section VI) and conclude the paper (Section VII).

Manuscript received November 9, 2010. Manuscript accepted for publication January 15, 2011.

The authors are with the University of Nantes / LINA (CNRS - UMR 6241), 2 rue de la Houssinière, B.P. 92208, 44322 Nantes Cedex 3, France (e-mail: first.last@univ-nantes.fr).

II. Related Works

The methods to handle the task we address depend on the granularity of the derivation and the transformations involved [1]. Texts that wholly derive from another one are better identified with suffix trees and string alignment methods [2], or using chunks frequency models when rewritting is involved [3]. Texts partially derived are better identified using matching chunks [4].

The n-grams overlap approach usually gives the best results for moderatly rewritten partially derived texts [4], [5]. It has been generalized and formalized by [6] as *w-shingling*. It consists of counting the contiguous subsequences of tokens (w-shingles) two texts have in common using a set theory based similarity metric. The assumption is that the more w-shingles the texts have in common the more probably they derive from each other. The set of the w-shingles of a text is its *signature*. The tokens composing the w-shingles can be any textual elements corresponding to a particular description. A *descriptor* describes the nature of these tokens as well as how they are combined into w-shingles. Several descriptors have been experimented in the litterature: fixed-length characters chunks [7], hashed breakpoints [8], words n-grams [6], [4], [5], sentences [9].

The major limit of this signature approach is its cost. The generated signatures are as large as the text which is inapropriate to handle large amount of data. For example, as word n-grams are not linguistically anchored, the signatures using this descriptor must contain all the overlapping n-grams of a text to match modified texts. This results in a signature even larger than the text itself, impacting the storage and the computational costs. One solution is to hash the tokens and only consider some meaningful bits of the hash therefore reducing the size of the fingerprint [10], [11]. However, the link to the elements in the texts are lost which is acceptable for near-duplicates as the whole document is derived but may not be for other kind of duplicates. We propose to focus on the choice of descriptors that are less numerous in the texts but are more effective at identifying derivations.

III. Building a Corpus to Evaluate Derivation Detection

A crucial question with NLP studies is the availability of a corpus resource with the wanted language phenomenon annotated in order to be able to infer and to test some hypothesis to retrieve it.

In the domain of the derivation detection, a few corpora with such annotations are available in English (METER [12], NTF and NTF2 [13], PAN [14]), no such resource is currently available in French. We note two major trends in building derivation corpora: (i) artificially generate derivations from a collection of texts by mixing them together [14], (ii) manually retrieve existing derivatives (from the Web for example) [12] or ask human to create some [14]. Both methods offer advantages and drawbacks. On the one hand, the artificial approach allows to quickly get a resource by performing automatically morphological, lexical, syntactic and semantic text edits (deletion, insertion, inversion, substitution) at various degrees and text granularities. The main drawback of this approach is that there is no mean to evaluate how much these transformations stand for natural language and consequently potential derivations. On the other hand, the major advantage of manually writting or searching for existing derivatives is that it may lead to get actual instances of a derivation process. Its drawbacks are that it needs time and fund to build a substantial corpus by searching derivatives and futhermore, it is often impossible to systematically control the search space as well as to be sure about the existence of the derivation links.

We argue that another way of building quickly a substantial corpus with actual derivation relations between documents is to use available corpora which include the annotation of some actual transformations between the documents, such as summarization synthesis, translation, revisions... As the manual simulation of the derivation process, this approach may not cover all the potential types of derivatives but the process to acquire them will be faster and probably cheaper. In this paper, we worked with a corpus made of revision texts.

Working with revisions is interesting for several reasons: the revision is a well-controled derivation type (sources and derivatives are easily identifiable, the derivation degree can be measured by the number of revision), it includes various forms of transformations such as spelling and grammar errors correction, insertion and deletion of contents, rephrasing,... We chose to work with *Wikinews* which is a project of the Wikimedia Foundation. Based on the idea of a collaborative journalism, Wikinews is a multilingual free-content[1] news source wiki. In addition to a head version of a news article, revisions and potential translations of the news are also available. We built our corpus from the data export of the French version of Wikinews[2] in date of November the 13^{rd} 2009. All the news articles having more than 10 revisions were selected; this constraint was set in order to reinforce the probability of getting suspicious texts with high degrees of edit operations from an initial source text.

The corpus is structured like the PAN corpus. It distinguishes the source texts and their derivatives. We choose to consider the first version of a news article as the source text and all the following revisions as the derivatives. As a matter of fact, the roles of not-derivative texts of a given source are played by the derivative texts of all other source texts. Since the PAN corpus is currently the reference to hold the evaluation of a derivation detection task, we adopted its file format conventions in order to ensure compatibility with it. The corpus is made of 221 source texts and 2,670 derivatives. On average a news article contains 604 words.

[1]Released under *Creative Commons Attribution 2.5*

[2]The Wikinews dumps can be downloaded from http://download.wikimedia.org. We used the *UIMA mediawiki engine* (http://code.google.com/p/uima-mediawiki-engine) to select and extract the raw texts from the news files.

IV. Approach

We address the task of detecting the derivatives of a given source. We particularly focus on document level derivatives, *i.e.* texts whose content is mainly derived from the source text as opposed to texts where only some minor passages are derived from the source text. Our goal is to develop a low operational costs method of detection.

As discussed in Section II, for a signature method to be operational, we must reduce the number of its elements. In order to do so, we must find more effective descriptors than word n-grams. In our opinion, this effectivity is a consequence of the specificity and the invariance of the descriptor. The idea that underlies the *specificity* is that a match on a signature element is more worth it if this particular element is only found in the source text that if it is a common element found in almost any text[3]. In other words, the less common a descriptor is the better it will discriminate the document. The *invariance* represents the ability of the descriptor instances to be preserved by the derivative process. In other words, the concept or the reference introduced by the instance should be found in the source and its derivatives.

In this paper we explore the use of descriptors chosen for their specificity or invariance: hapax n-grams, named entities and nominal compounds. We also explore their combination as pros of each may overcome cons of the others.

A. Hapax n-grams

The hapax n-grams both extend the idea of using word n-grams while implementing the principle of specificity and reducing the number of elements in the signature. Moreover, they can be easily extracted using a reference distribution.

Hapax n-grams are a great example of specificity. They extend the concept of w-shingling by reusing word n-grams as basic units composing the signature, so their implementation is not much different that the w-shingling method. However, a filtering step is necessary as we only keep extremly specific elements : these appearing only once, the hapax. More precisely, we select from the word n-grams of a text the ones with a df (document frequency) of one or less given a reference distribution. The method hopefully reduces the

[3]It is a direct interpretation of the fact that the more an element derive from the Poisson distribution the more it is useful to discriminate the hidden relationships behind text [15]

number of elements in the signature as it is a filtered version of the original w-shingling.

The only difficulty in building such a signature is to obtain a reference distribution. The reference distribution must be computed over a corpus of the same genre and same language as ours. Using the same corpus is not an option as it would result in an identical distribution while we are interested in variations, and as it is mainly made of derivatives it is not representative of the language (redundancy of reused expressions). This would lead to erroneous results. Instead we use the pages of Wikinews that are not part of the corpus. We only keep one revision per article to avoid derivatives, we especially select the last revision as it is generally the longest and the most correct. The resulting corpus is composed of 1,027 French press articles, representing 289,288 words. The word n-grams are extracted from this reference corpus and stored in an index with their df. Therefore, we considered as hapax the n-grams of our corpus with a $\mathrm{df} <= 1$ in this index.

B. Nominal Compounds and Named Entities

So far, researchers payed relatively little attention to linguistic-based descriptors. According to us, signatures based on some linguistically motivated descriptors can enhance the detection performance compared to n-grams w-shingling signatures.

First, since a linguistic descriptor is defined by some grammatical and semantic constraints, its instances are a subset of the text which is a solution to reduce the size of the signature Second, some linguistic descriptors may be considered to be more relevant than others to describe the content of a document. Among them, we include the nominal compounds and the named entities. Third, since instances of these descriptors result from a linguistic choice of the author, they provide a greater probability to integrate specificities from the author of the source text.

We decide to consider two distinct categories of linguistic descriptors: the named entities (names of persons, organisations, locations) and the nominal compounds. We assume that if the instances of these descriptors from a source are found in a suspicious text they enhance the probability for the suspicious text to be a derivative.

We choose to observe the named entities because they usually designate the referents of the actors or of the context elements of the events reported in news. For named entities extraction, we used the French system Nemesis [16]. Nemesis follows a lexical and grammar-based approach with some automatic learning techniques to enrich the lexicon. It achieves a performance of 95 % in precision and 90 % in recall for recognizing anthroponyms and toponyms in press texts.

Whereas the named entities constitute expressions which stand for referents, the nominal compounds are generally used as the most syntactically plausible class of terminological candidates to model the concepts of the knowledge domains. They constitute more than 80 % of the domain specific terms for the specialized languages [17], but they are also used in the informal language. We use the grammar-based patterns[4] proposed by [18] to extract the nominal compounds: N A (*emballage biodégradable*, *protéine végétale*), N (P (D)) N (*ions calcium*, *protéine de poissons*, *chimioprophylaxie au rifampine*), N à Vinf (*impôts à acquitter*, *fonds à venir*). These patterns are recursive and may admit some variations such as N N A (*forces armées britaniques*), N A (P (D)) N (*lait cru de brebis*) or N (P (D)) N A (*protéine d'origine végétale*, *réunion de la Commission Parlementaire*). In our implementation, we only considered the precited patterns and variants without further recursive variations. Overlapping nominal compounds retrieved by different patterns were allowed in order to enhance the capability of detecting partial rewriting. We used the Apache UIMA Tagger[5] which was trained on the French treebank [19] to compute parts-of-speech on the texts.

V. Evaluation Protocol

The systems to detect derivatives are usually evaluated as classifiers using the computed similarity scores, whether they categorize pairs of documents [4] or pairs of passages [14]. Usually, the classifier is based on a simple similarity threshold which differs derivatives from not-derivatives. Thus, the underlying comparison method is not evaluated as the focus is on the correct distribution of pairs in their respective classes. This kind of evaluation is appropriate for a decision making system which we believe is not a relevant choice for our problem.

We think derivatives detection systems should be seen as decision support systems and evaluated as such. Therefore, the evaluation must measure how the system sorts out relevant candidates and help a human to take a decision regarding the derivative status of a text by providing relevant insights. In the continuity of the works from [20] and [21], we think an IR-like evaluation is the best choice for such a system. We sort pairs of documents (one source and one suspicious) according to their similarity score computed by the system. The highest scores obtain the highest ranks.

We are interested in three evaluation axes: the quality of the ranking (pairs with derivatives should obtain the highest ranks and not-derivatives the lowest), the discrimination capability of the system (derivatives scores should be very different from not-derivatives ones) and the computation costs (storage cost and execution time of the system). We also present the results of the w-shingling approach that we use as a baseline.

A. Quality of the Pairs Ranking

The quality of the pairs ranking is the most obvious property to evaluate the quality of our system. It is comparable to the precision and recall measures for the evaluations as

[4]The Part-Of-Speech tag A stands for Adjective, N for Noun, D for Determiner, P for Preposition and Vinf for Infinive Verb. *à* is a specific preposition.

[5]http://uima.apache.org/downloads/sandbox/hmmTaggerUsersGuide/

classification tasks in the sense that it evaluates how well are derivatives identified.

The mean average precision (MAP) metric is well suited to measure the quality of the ranking as it combined precision and recall like notions without the need for a binary categorization between derived and not-derived. It is the average of the regular precision metric computed over a growing window of N ranks starting from rank 1 (Equation 1). We use for N the rank of the last derivation link in the ranking. The recall is expressed here through the denominator N: the highest N is the more there are not-derivatives before the last derivative and the more important is the impact on the MAP.

$$\begin{aligned} &\text{MAP} = \frac{\sum_{r=1}^{N} P(r)}{N} \\ &r \text{ a rank} \\ &N \text{ the highest rank considered for the computation} \\ &P(r) \text{ the precision computed over rank 1 to } r \end{aligned} \tag{1}$$

B. Discrimination Capability

The discrimination capability of the method reflects how well the method makes a difference between derivation links and not-derivative ones.

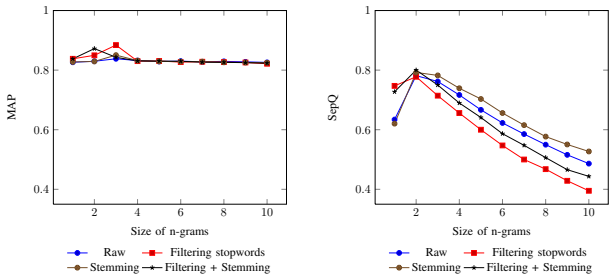
This property of the system is measured as the size of the buffer between derivation links similarity scores and not-derivative ones with the asumption that the larger this buffer is, the more each link is considered differently from the other by our system. The separation is the difference between the similarity score of the highest not-derivative in the ranking and the lowest derivative one. We introduce the SepQ that, instead of using the extrems of each, considers the similarity scores of the third quartile of the derivation links and the first quartile of the not-derivative ones (Equation 2). Indeed, the consideration of a unique individual, in addition an extrem, may not reflect the group. Therefore we prefer to measure the distance between the most significative $\frac{3}{4}$ of each group: the highest similarities for the derivation links and the lowest for the no derivation ones.

$$\begin{aligned} &\text{SepQ} = s_{deriv} - s_{\neg deriv} \\ &s_{deriv} \text{ similarity score of the } 3^{rd} \text{ quartile} \\ &\text{of the derivatives} \\ &s_{\neg deriv} \text{ similarity score of the } 1^{st} \text{ quartile} \\ &\text{of the not-derivatives} \end{aligned} \tag{2}$$

C. Computational Costs

As our goal is to develop a low operational costs method of detection, we want to measure the computational cost of the system.

The system processes in two steps: the extraction of the signature and the comparison of the signatures pairs. The former is done only once so its impact on the computational costs can be neglected compare to the latter. For our approach, the complexity of the comparisons between two signatures is dominated by the computation of the intersection of the signatures which itself is linear with the size of the signatures ($O(2|s|)$). Therefore, we measure the computational costs by measuring the size of each type of signature.



(a) MAP (b) SepQ

Fig. 1. Results obtained for various size of n-grams.

D. Baseline Approach

We use the w-shingling approach with word n-grams and the c_{max} similarity metric (*Equation 4*) as a baseline. Experimentations not reported here show that this symetric measure (*Equation 3*) gives better results than the classic containment metric (*Equation 3*) for our corpus.

$$\begin{aligned} &c(a,b) = \frac{|\Pi(a) \cap \Pi(b)|}{|\Pi(a)|} \\ &\Pi(d) \text{ the w-shingling of document } d \end{aligned} \tag{3}$$

$$c_{max}(a,b) = \max(c(a,b), c(b,a)) \tag{4}$$

We explore several parameters regarding the n-grams to obtain the best possible results for the baseline. Thus, we experiment several sizes of n-grams as well as some morphological (stemming) and lexical (stopwords removal) normalizations. The results of these variations are presented in Figure 1. The MAP (Figure 1(a)) is globally constant independantly of the size of n-grams with just two peaks: stopword filtered 3-grams and stopwords filtered and stemmed 2-grams. The SepQ curve (Figure 1(b)) has a totally different shape as results fall with the increasing size of n-grams. The maximum is reached for 2-grams, whatever the type of n-grams. With regard to these results, the stopwords filtered and stemmed 2-grams is the configuration we choose as our baseline. The measured MAP for this configuration is of 0.872 and the SepQ of 0.800. We will consider the size of the corresponding signature as a tare for the next methods.

The Wikinews corpus represents a particular kind of derivations: revisions of press articles in French. The independance of the MAP relative to the size of n-grams is because of the sparse modifications. The revisions globally cover each others, but the raw n-grams are not adapted to capture the variations. The best results are obtained with some normalization and small n-grams. We think that the normalization removes unstable parts especially the endings associated with gender and number, while the small n-grams (2-grams and 3-grams) capture some stable

syntaxic constructions. This particular role of small n-grams is somehow supported by the SepQ results.

VI. Results

Table I presents the results of the different descriptors described in Section IV and their combination. Results are compared to these of the baseline. In the last section, we discuss the results we obtain for the linguistic descriptors by manually looking at some selected pairs of compared texts.

A. Results for Each Descriptor

We measure the performance of hapax n-grams with n varying from 1 to 10. The MAP of the 2-grams baseline outperforms all the MAP scores of the individual descriptors, *a fortiori* the hapax MAP score. We note that for $n = 1$ and $n = 3$ the hapax MAP gives a better result and for $n >= 4$ they are quite similar. The SepQ is roughly the same whatever n is. The value is decreasing while n is increasing. In Table I, we present only the best results which are obtained with 2-grams and 1-grams respectively for the MAP and the SepQ. As a general trend, the MAP scores of the different descriptors never outperform the baseline but the SepQ ones are better and the corresponding signatures are smaller.

More precisely, Table I indicates that the MAP score of the named entity descriptor decreases of 0.22 points while the discrimination capability increases of 0.03 points. The most interesting observation we note concerns the signature size which corresponds to a significant decrease of the baseline signature size (5 % of this latter). Indeed, this decrease impacts positively the signatures storage cost and so the cost of the signatures comparison.

Turning now to the nominal compound descriptor, we can see in Table I that it gives lower scores than the baseline. Indeed, the MAP of the nominal compound descriptor results in a slight decrease of 0.04 points and its discrimination capability in also a slight decrease of 0.06 points. However, while these results are slightly lower, in comparison there is again a significant decrease of the signature size (15 % of the baseline).

B. Combination of the Descriptors

The combination of descriptors can be considered at different stages: at the signature building stage by combining all signatures as one or at the similarity measure stage by a simple linear combination. In this paper, we choose to perform the latter for at least two reasons: first, it makes the signature building process easier allowing to compute separately each descriptor signature. Second, it easily allows to control the weight of each descriptor in the combination.

We define the linear function, $\mathrm{sim}_{comb}^{a,b,c}$, to combine the similarity scores we obtained by the differents approaches such that:

$$\begin{aligned}\mathrm{sim}_{comb}^{a,b,c}(t_1,t_2) = a.\,\mathrm{sim}_H(t_1,t_2) + b.\,\mathrm{sim}_{NE}(t_1,t_2)\\ +c.\,\mathrm{sim}_{NC}(t_1,t_2)\\ t_1, t_2 \text{ two texts on focus}\\ \mathrm{sim}_H, \mathrm{sim}_{NE}, \mathrm{sim}_{NC} \text{ scores of the } H\text{apax,}\\ \quad N\text{amed } E\text{ntity and } N\text{ominal } C\text{ompound methods}\\ a, b, c \text{ coefficients}\end{aligned} \tag{5}$$

We experimented a range of values from 0 to 3 for each coefficient. As shown in Table I, the combination $\mathrm{sim}_{comb}^{2,1,1}$ outperforms the baseline performances. Moreover, the various combinations reported all outperform the results of their individual constituent. This shows that being able to set correctly the coefficient of each descriptor can improve the combination results. Eventually we note that any combination has a significantly lower signature size than the baseline.

C. Discussion

In order to discuss qualitatively the results we obtain with linguistically motivated descriptors, we manually observed some compared texts: the pairs of texts with no actual derivation link but with the highest similarity rankings and the pairs of texts with an actual derivation link but with the lowest similarity rankings[6].

We found three main reasons why some pairs of texts with no actual derivation link have a high similarity ranking. One reason is attributed to the comparison of signatures with very different size. As a consequence, the more elements a signature has the higher the probability is that this signature includes some elements of the compared signature. Another reason of potential high similarity ranking is due to a low quantity of descriptor instances in the compared texts. This was specifically observed for the named entity descriptor. In general the texts we processed use at most half a dozen of distinct named entities. As a result, one single shared element has strong impact in the score similarity measure. In addition to these remarks, we observed that some of the shared elements between the signatures belongs to a common lexicon which artificially increases the score of similarity. This was the case for the named entities descriptor with common toponyms such as *France*, *United States*, *North*... and also the case for the nominal compounds descriptor with for example some terms related to the model of the document such as "*source*" or "*exclusive right*".

For the named entities descriptor, we found one main explanation about why some pairs of texts with an actual derivation link got a low similarity ranking. Mainly, this was due to the fact that the shared elements were insignificant regarding the signature size. This observation is reinforced by the text variation of the named entities. Indeed *the President of the French Republic* and *the President* count for distinct elements in the signature and do not match if they are

[6]Pairs of texts with a null similarity score were not considered.

TABLE I
COMPARISON OF THE DESCRIPTORS SCORES IN TERMS OF MAP SCORE, SEPQ SCORE AND SIZE RELATIVELY TO THE BASELINE SIGNATURE SIZE. FOR EACH SCORE, WE SKETCH ITS EVOLUTION COMPARED WITH THE BASELINE: ↗ INDICATES A SCORE INCREASE, ↘ A SECREASE AND = EQUIVALENT SCORE

Descriptor(s)		MAP		SepQ		Signature size	
Baseline		0.872		0.800		100 %	
Hapax	max(MAP): 2-grams (H2)	0.856	↘	0.807	↗	78 %	↗
	max(SepQ): 1-grams (H1)	0.849	↘	0.866	↗	9 %	↗
Named entities (NE)		0.646	↘	0.833	↗	5 %	↗
Nominal compounds (NC)		0.831	↘	0.746	↘	15 %	↗
$1 \cdot \mathrm{sim}_{NE} + 1 \cdot \mathrm{sim}_{NC}$		0.846	↘	1.242	↗	20 %	↗
$2 \cdot \mathrm{sim}_{H1} + 1 \cdot \mathrm{sim}_{NE} + 1 \cdot \mathrm{sim}_{NC}$		0.875	↗	2.906	↗	28 %	↗
$1 \cdot \mathrm{sim}_{H2} + 2 \cdot \mathrm{sim}_{NE} + 0 \cdot \mathrm{sim}_{NC}$		0.872	=	1.987	↗	93 %	↗

compared. For the nominal compounds descriptor, this result was due to an intrinsic property of the corpus. Indeed, it seems that some revisions of a piece of news were a translated version. This probably comes when an article was translated from a foreign language. As a consequence, despite the fact they point the same concepts, nominal compounds couldn't match.

VII. CONCLUSION

This paper has given an account of our work to build a derivation corpus, to set an appropriate evaluation protocole and eventually to evaluate some original descriptors. We believe that the methods we used can open up new paths for the studies of derivation detection. We provide a derivation corpus with revision relation for press texts which constitutes a concrete contribution to the scientific community since no resource were available for studying derivation in French. In addition thanks to an inherent property of the corpus source, it can be extended to include translation derivations. It is freely available and can be easily integrate to the PAN corpus because of its file format compatibility. Concerning our results, we show that descriptors such as 1-gram hapax and nominal compounds can provide a substantial gain in terms of signatures storage and comparison costs with only a slight loss of general performances. Our manual analysis shows that in regard to the text material of a news, these linguistic descriptors can play an important role to discriminate or characterize a text but their impacts remain quite sensitive to the size of the compared texts. Further research should investigate the temporal expressions (dates, times) and the numerical expressions (quantities, monetary values, percentages) as well as the named entities and the nominal compounds variations to enhance the capibility of these descriptors. In addition, more research needs to be undertaken to see whether it is possible to filter the common lexicons, by $\mathrm{tf} \cdot \mathrm{idf}$ for example.

REFERENCES

[1] S. M. Z. Eissen and B. Stein, "Intrinsic plagiarism detection," in *Proceedings of the 28th European Conference on IR Research (ECIR 2006)*, 2006, pp. 565–569. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.110.5366

[2] A. Aizawa, "Analysis of source identified text corpora: exploring the statistics of the reused text and authorship," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, vol. 1, 2003, pp. 383–390. [Online]. Available: http://portal.acm.org/citation.cfm?id=1075145

[3] N. Shivakumar and H. Garcia-molina, "Building a scalable and accurate copy detection mechanism," in *Proceedings of the 1st ACM International Conference on Digital Libraries (DL 1996)*, 1996, pp. 160–168. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.6064

[4] P. Clough, "Measuring text reuse," Ph.D. dissertation, University of Sheffield, mar 2003.

[5] C. Lyon, R. Barrett, and J. Malcolm, "Plagiarism is easy, but also easy to detect," *Plagiary*, vol. 1, pp. 1–10, 2006.

[6] A. Z. Broder, "On the resemblance and containment of documents," in *Compression and Complexity of SEQUENCES 1997*, 1997, pp. 21–29. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.779

[7] N. Heintze, "Scalable document fingerprinting (Extended abstract)," http://www.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html, 1996. [Online]. Available: http://www.cs.cmu.edu/afs/cs/user/nch/www/koala/main.html

[8] U. Manber, "Finding similar files in a large file system," in *Proceedings of the USENIX Winter 1994 Technical Conference*, October 1994, p. 1–10. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.12.3222&rep=rep1&type=pdf

[9] S. Brin, J. Davis, and H. Garcia-molina, "Copy detection mechanisms for digital documents," in *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD 1995)*, 1995, pp. 398–409. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.8485

[10] M. Henzinger, "Finding near-duplicate web pages," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '06*, E. N. Efthimiadis, S. T. Dumais, D. Hawking, and J. e. Kalervo, Eds. ACM, 2006, p. 284. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1148170.1148222

[11] Y. Bernstein, M. Shokouhi, and J. Zobel, "Compact features for detection of near-duplicates in distributed retrieval," in *Proceedings of the Symposium on String Processing and Information Retrieval*, 2006, pp. 110–121. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.3243

[12] R. Gaizauskas, J. Foster, Y. Wilks, J. Arundel, P. Clough, and S. S. L. Piao, "The meter corpus: a corpus for analysing journalistic text reuse," in *Proceedings of the 2001 Corpus Linguistics Conference*, 2001, pp. 214–223. [Online]. Available: http://nlp.shef.ac.uk/meter/

[13] H. Yang, "Next steps in near-duplicate detection for erulemaking," in *Proceedings of the 7th Annual International Conference on Digital Government Research (DG.O 2006)*, 2006, pp. 239–248. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.3732

[14] M. Potthast, B. Stein, and P. Rosso, "An evaluation framework for plagiarism detection," in *Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010*, 2010.

[15] K. W. Church and W. A. Gale, "Inverse document frequency (IDF): A measure of deviations from poisson," in *Proceedings of the Third Workshop on Very Large Corpora*, 1995, p. 121–130.

[16] N. Fourour, E. Morin, and B. Daille, "Incremental recognition and referential categorization of french proper names," in *Proceedings of the Third International Conference on Language Ressources and Evaluation (LREC 2002)*, vol. 3, 2002, pp. 1068–1074.

[17] F. Cerbah, "Exogeneous and endogeneous approaches to semantic categorization of unknown technical terms," in *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 2000, pp. 145–151.

[18] B. Daille, "Conceptual structuring through term variations," in *Proceedings ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, 2003, pp. 9–16.

[19] A. Abeillé, L. Clément, and F. Toussenel, *Building a treebank for French*. Kluwer Academic Publishers, 2003, pp. 165–187.

[20] T. C. Hoad and J. Zobel, "Methods for identifying versioned and plagiarised documents," *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 203—215, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2680

[21] D. Metzler, Y. Bernstein, B. W. Croft, A. Moffat, and J. Zobel, "Similarity measures for tracking information flow," in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM, 2005, pp. 517–524.

Assesing the Feature-Driven Nature of Similarity-based Sorting of Verbs

Pinar Öztürk, Mila Vulchanova, Christian Tumyr, Liliana Martinez, and David Kabath

***Abstract*—The paper presents a computational analysis of the results from a sorting task with motion verbs in Norwegian. The sorting behavior of humans rests on the features they use when they compare two or more words. We investigate what these features are and how differential each feature may be in sorting. The key rationale for our method of analysis is the assumption that a sorting task rests on a similarity assessment process. The main idea is that a set of features underlies this similarity judgment, and similarity between two verbs amounts to the sum of the weighted similarity between the given set of features. The computational methodology used to investigate the features is as follows. Based on the frequency of co-occurrence of verbs in the human generated cluster, weights of a given set of features are computed using linear regression. The weights are used, in turn, to compute a similarity matrix between the verbs. This matrix is used as an input for the agglomerative hierarchical clustering. If the selected/projected set of features aligns with the features the participants used when sorting verbs in groups, then the clusters we obtain using this computational method would align with the clusters generated by humans. Otherwise, the method proceeds with modifying the feature set and repeating the process. Features promoting clusters that align with human-generated clusters are evaluated by a set of human experts and the results show that the method manages to identify the appropriate feature sets. This method can be applied in analyzing a variety of data ranging from experimental free production data, to linguistic data from controlled experiments in the assessment of semantic relations and hierarchies within languages and across languages.**

***Index Terms*—Verb features, verb sorting, similarity.**

I. Introduction

SORTING tasks are a popular knowledge elicitation technique used in psychology and cognitive studies [1], [2]. In a typical sorting task participants are asked to sort in groups items in a particular domain. This kind of task rests on the common assumption that, in categorization processes, humans rely on specific features that differentiate one group of objects from another, and that these features characterize and define the group in a broader domain [3].

We designed a sorting task to study the semantic domain of verbs of human locomotion below the basic level ([4], [5], [6]). Specific verbs of locomotion include words, such as English *strut*, *stroll*, *gambol*, *hop*, and the like.

Our main assumption is that the way speakers group those verbs is revealing about the semantic structure of this field. Our hypothesis is that the size (how many) and constitution (what verbs) of these groups can be used to derive the semantic features that characterize both individual lexical items and the domain as a whole. We investigated whether and how it is possible to discover such relations and patterns for the set of motion related verbs, based on verb clusters provided by the human subjects. The paper presents a computational method that aims to discover the most salient features and their degree of saliency.

The approach adopted in this paper resembles vector-based semantic space models which rely on patterns of word co-occurrence to derive similarity estimates ([7], [8]). The difference from such approaches is that they aim to extract information either from the broader lexical or from the syntactic context of the target word, while our approach targets groupings based on closer semantic similarity within a well-defined conceptual and semantic domain (e.g., words describing human locomotion). In our formalisation, both the columns and the rows in the raw matrix are target words, i.e. it is a verb-verb matrix. Even though this approach might appear narrow and highly restricted to the domain it applies to, it is justified on the basis of research and intuitions in lexical semantics, as well as human categorization. Thus, studying the grouping of words that are partially synonymous with each other and can be subsumed under the same superordinate term, can be used to reveal the underlying features that characterize this semantic field and the basic (superordinate) term. Moreover, Semantic space models have been criticized exactly on the grounds of not being able to address the nature of the semantic relationship that underlies proximity of words in the semantic space [7]. We address this shortcoming by using a feature-verb matrix to estimate the weighting of features.

Another difference between the current approach and existing approaches in cognitive science and psychology is that, while the latter have used human elicitation to verify the findings from semantic space models [9], we adopt a parallel experimental strategy: we seek to find out the extent to which a computational model based on human data can improve by using featural data elicited from the human data.

The outline of the paper is the following. We first introduce the human sorting task experiment and its linguistic background in the next section. We then proceed, in section III with the computational method for computing feature weights and the clusters based on various combinations of the features.

Manuscript received November 16, 2010. Manuscript accepted for publication January 22, 2011.

The authors are with the Norwegian University of Science and Technology, Trondheim, Norway (e-mail: Pinar.Ozturk@ifi.unit.no).

Section IV presents the computational experiments and the results of applying the computational method. We discuss the obtained results in section V and conclude with a summary and future directions in section VI.

II. Human Experiments

Germanic languages are characterized by a rich system of specific verbs describing locomotion, and the distinctions among the items in this domain are not always very clear. Furthermore, little is known about the way native speakers of these languages acquire such highly specific vocabulary, and whether they use salient perceptual features of the actions these words denote, and then map these features onto the lexical items at hand or simply rely on the linguistic contexts in which they first encounter these verbs [10].

As a first step in studying the native speakers's knowledge of specific locomotion verbs, we asked native speakers of Norwegian to group 41 verbs that were selected through a 3 step process, a semantic recall task, an elicitation task, with results from both being compared to a comprehensive list compiled on the basis of dictionary information [6].

The verbs appeared on small paper cards and participants were asked to sort them in groups by similarity. Participants are then asked to describe what features they have used in the grouping process. All the features mentioned by one or several subjects constitute the candidate feature set including 15 features. Using the computational method described in the next section, we tried to select the subset from this candidate set of the features that were most influential in the overall sorting experiment.

To avoid confounding of the results, the human subjects were given the opportunity of placing verbs whose meaning they did not know or, for some reason, whose placement they felt uncertain about, in a separate group labeled "out", which indicated exclusion from the sorting. Verbs excluded in this way are considered as a negative contribution and were excluded from further analyses. A total of three verbs were excluded by more than two subjects and were removed from the dataset for analysis.

The groups for each participant were photographed by digital camera, and the results for all participants were manually entered in an excel file and consequently converted into a *verb co-occurence* matrix of which each cell indicates how many of the subjects put the two corresponding verbs into the same group. These raw data served as the input for agglomerative hierarchical clustering. This matrix constitutes also the input to the computational method described in the next section.

III. The Computational Method

The inputs to the method are a feature-verb matrix representing subjects' description of which features were taken into consideration when grouping verbs, and the verb co-occurence matrix prepared after the human experiment. In this paper, the feature-verb matrix has size 15 x 41, while the verb co-occurence matrix is of size 41 x41.

The overall method is summarized in Algorithm 1 where S_{human} is a verb-verb matrix, i.e., the co-occurence matrix generated by accumulating the sorting data provided by the subjects. $S_{human}(v_i, v_j)$ represents the number of subjects who put verbs v_i and v_j into the same group. It is considered to represent the human judgment of similarity between the verbs. S_{comp} is the computed (more precisely, to be computed) feature-based verb similarity matrix.

Algorithm 1 Method

1: $C_{human} \leftarrow$ Cluster data based on S_{human}
2: Matrix A $\leftarrow$ human description of feature-verb relations
3: **repeat**
4: Compute feature weights $\mathbf{W}$ (as described in algorithm 2)
5: Generate weighted feature-based verb similarity matrix S_{comp} using $\mathbf{W}$ and A (details described in algorithm 3)
6: $C_{comp} \leftarrow$ Cluster the data based on S_{comp}
7: Evaluate alignment between C_{human} and C_{comp}
8: **if** $C_{comp} \not\simeq C_{human}$ **then**
9: Remove the feature with the lowest weight
10: **end if**
11: **until** $C_{comp} \simeq C_{human}$ **or** # of features < 2

The algorithm describes the process of evaluating the calculated feature weights with regard to the grouping data provided by the human subjects. The grouping data are clustered (the result is denoted as C_{human} in Algorithm 1) using agglomerative hierarchical clustering. After the weights of the features are computed as explained in section III-A, a weight-based verb similarity matrix S_{comp} is computed (explained in section III-B) using these weights. Then, the verbs are clustered again using the same clustering methods, this time using the new similarity matrix S_{comp}. These clusters are depicted as C_{comp} in Algorithm 1.

If the computed clusters C_{comp} and human based clusters C_{human} align, i.e. are fairly similar (depicted as $C_{comp} \simeq C_{human}$), the features and weights are considered to indicate what the human subjects based their clustering of the verbs on. If the clusters do not align, some features are removed from the set of features and C_{comp} is computed anew, and the process is repeated until an alignment has been achieved.

A. Computation of Weights

A central idea underlying the proposed method is that similarity between two verbs is equal to the weighted sum of the similarities between the involved features, which is defined by Equation 1 where w_n is the weight of feature a_n.

$$S(v_i, v_j) = w_1 f(a_{1i}, a_{1j}) + w_2 f(a_{2i}, a_{2j}) + \cdots + w_n f(a_{ni}, a_{nj}) \quad (1)$$

Values $f(a_i, a_j)$ represent the similarity between verbs v_i and v_j computed applying the similarity metric *f* on the cells of the feature-verb matrix *A* which captures subjects' description of which features were used when placing each verb in a group. The f function uses one of the well-known similarity measures for binary vectors [11].

In addition to the rationale captured by Equation 1, Equation 2 conveys another central assumption in our method:

$$S_{comp}(v_i, v_j) = S_{human}(v_i, v_j) \tag{2}$$

The instantiation of equations 1 and 2 for all verbs yields the following linear system of equations, which, when solved, provide values for the weights $w_{1...n}$ for the features.

$$\begin{bmatrix} f(a_{11}, a_{12}) \\ f(a_{11}, a_{13}) \\ f(a_{11}, a_{14}) \\ \vdots \\ f(a_{nm}, a_{nm-1}) \end{bmatrix} . \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} S_{human}(v_1, v_2) \\ S_{human}(v_1, v_3) \\ S_{human}(v_1, v_4) \\ \vdots \\ S_{human}(v_m, v_{m-1}) \end{bmatrix}$$

Algorithm 2 describes the process of calculating the weights of the 15 features in the candidate feature set. It uses the feature-verb matrix *A* and the human generated verb co-occurrence matrix S_{human} to calculate the weights; both are data from human experiments. The value of feature a_n for a verb v_j is denoted as a_{nj} and is found in the feature-verb matrix *A*. The similarity of feature a_n between verb v_i and v_j is computed by $f(a_{ni}, a_{nj})$, and w_n is the weight or importance of feature a_n. The value of the weights are determined by solving the set EQ of $\frac{i^2}{2}$ linear equations where m denotes the number of verbs. $S_{human}(v_i, v_j)$ denotes the number of subjects having placed the verbs i and j in the same group. A similar approach is taken in [12] where the concerned items are movies and similarity between two movies is associated with the number of persons who rated both of these movies.

Algorithm 2 Calculation of weights

```
n ← Number of features
EQ ← Empty set of linear equations
for each verb v_i do
    for each verbs (v_j), j = (i + 1) do
        Add w_1 f(a_1i, a_1j) + ··· + w_n f(a_ni, a_nj) = S_human(v_i, v_j) to EQ
    end for
end for
Solve EQ for W
return |W|
```

B. Computation of Feature-based Verb Similarity Matrix

Algorithm 3 describes the process of calculating the similarity between verbs based on the feature weights which were computed using algorithm 2. The similarity between two verbs i and j, denoted as $S_{comp}(v_i, v_j)$ is then computed using Equation 1. This process generates the feature-based similarity matrix S_{comp}.

Algorithm 3 Calculation of verb similarity based on weights

```
n ← Number of features
for each verb v_i do
    for each verb (v_j), j = (i + 1) do
        S(v_i, v_j) = Σ_{k=1}^{n} w_k f(a_ki, a_kj)
        S_comp(i, j) = S(v_i, v_j)
    end for
end for
return S_comp
```

IV. Experiments and Results

We have conducted a set of experiments to see how the different distance metrics would affect the clustering performance, and the effect of the different linkage methods in hierarchical clustering of the verbs. Another set of experiments were devoted to investigating which features are most salient in the clustering. For this purpose we used the algorithm 2 to determine the weights of features and algorithm 3 to compute the distance matrix. Then we applied hierarchical clustering, again using different linkage methods.

Regarding the human clustering, we have experimented with the distance metrics provided by MATLAB such as *Jaccard*, *Correlation*, *Euclidean*, *Minkowski*, *Cosine*, *Chebychev* etc. In addition, we have implemented the Multiset distance metric [1] which has proven appropriate in previous analyses of verb similarity (Dimitrova-Vulchanova et al., in press). As to linkage methods, MATLAB provides several methods including *Centroid*, *Median*, *Single*, *Average*, and *Complete*. The best clustering tree of human grouping data was found to be provided by *Euclidean* as the distance metric and *Average* as the linking method. Euclidian Average has proven useful in plotting cross-linguistic differences and similarities in the naming of cutting and breaking scenes in a representative sample of world's languages (Majid et al. 2008), and our results confirm the advantages of the method in similar tasks. Figure 1 illustrates Jaccard-Average combination while Figure 2 shows the cluster tree when Euclidean-Average combination is used.

We have identified a set of features to have a role, in various degrees, in the human grouping process (referred to as feature-verb matrix above). Our anticipation is heavily based on the descriptions provided by the subjects who participated in the experiments. However, we have also supplemented these

[1] Calculated as

$$d(S_i, S_j) = 1 - \frac{\sum_{o \in O} min(n_{0,S_i}, n_{0,S_j})}{\sum_{o \in O} max(n_{0,S_i}, n_{0,S_j})}$$

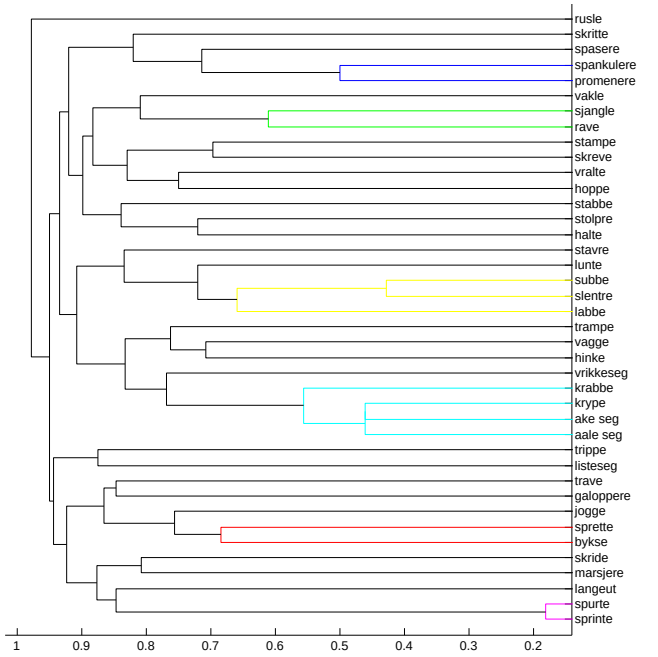


Fig. 1. Clustering of human grouping data using Jaccard metric and Average link.

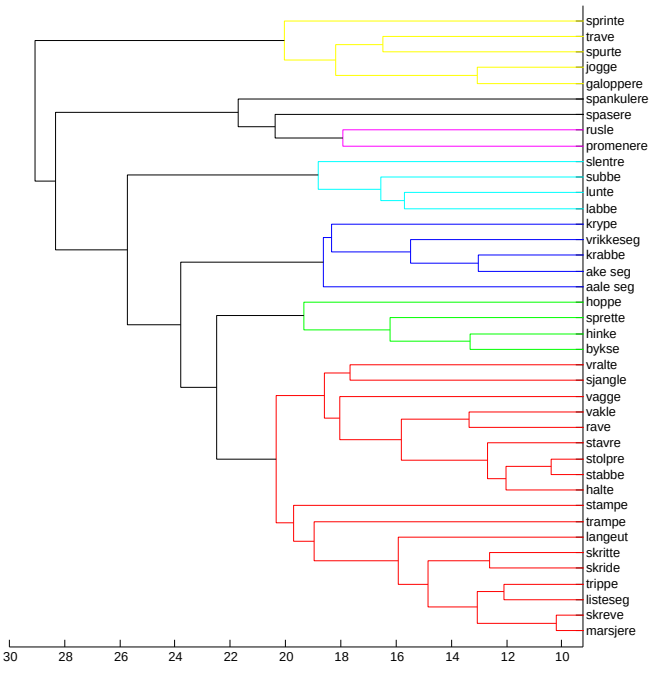


Fig. 2. Clustering of human grouping data using Euclidean metric and Average link.

data with information from the dictionary definitions of the verbs, as well as human expert judgments. Using the method presented in section III we have estimated the weights (i.e., salience) of the features in the grouping process. As a next step we computed the distance matrix (i.e, the verb-verb matrix) to be used as input for the clustering. In this process we have experimented with different distance metrics and linking methods, as already described above.

Initially we had 15 features: *contact* (with substrate), *limbs* (body parts involved in moving), *propulsion*(pattern), *position* (of parts of the body not involved in the motion), *symmetrical (motion pattern)*, *sideways* (motion pattern), *stride* (length), *(typical)* agent, *cause*, *sound*, *speed*, *effort*, *agility*, *social* (context), *purpose*. The computed weights of these features are shown in figure 3.

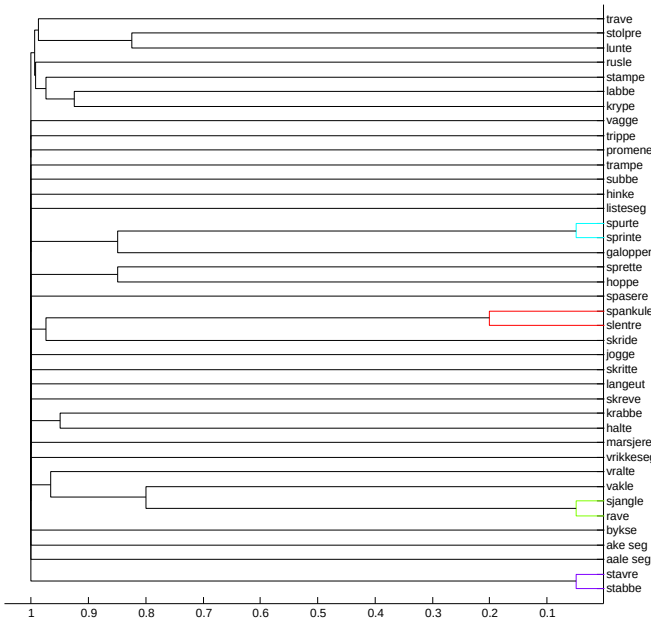


Fig. 4. Clusters based on 15 features, Jaccard metric and Average linkage was used.

Using these weights we studied the hierarchical trees of the verbs. The Euclidean-Average combination has shown the best performance, according to human expert judgments. Two of the hierarchical trees based on these weights are shown in Figures 4 (using Jaccard metric and Average linkage method) and 5 (Euclidean and Average). As can be seen in the sorted feature set according to weights (see Figure 3), some of the weight values are significantly lower than others. Moreover, both the Jaccard Average and the Euclidian Average clusters based on all 15 features were not particularly successful in capturing the structure of the semantic field and deviate substantially from the human data cluster, as judged by human experts. This deviation from the human data cluster may suggest that either a/some features are irrelevant, and/or b/ not all features capture adequately the semantic relationship in the semantic field at hand. Therefore we have analyzed different and fewer numbers of feature combinations. The feature weights showed the same trend, while clustering performance varied depending on the number of features and which features were chosen. In general removal of the two low-weight features *propulsion* and *position*, the two middle

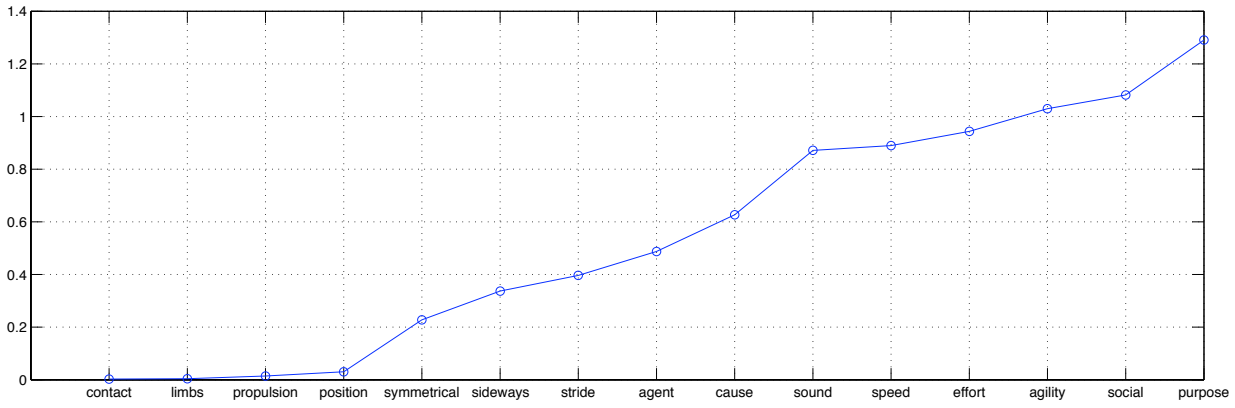


Fig. 3. Weight values of the 15 features.

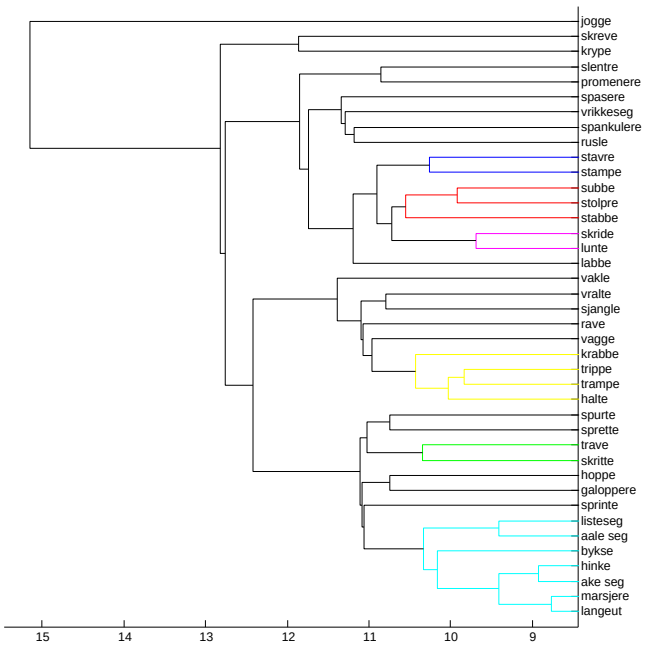


Fig. 5. Clusters based on 15 features, Euclidean metric and Average linkage was used.

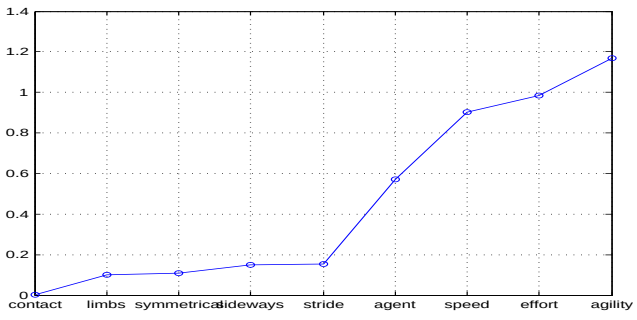


Fig. 6. Weights for 9 features.

features *cause* and *sound*, and the two high-weight features *social* and *purpose* produced a balanced effect.

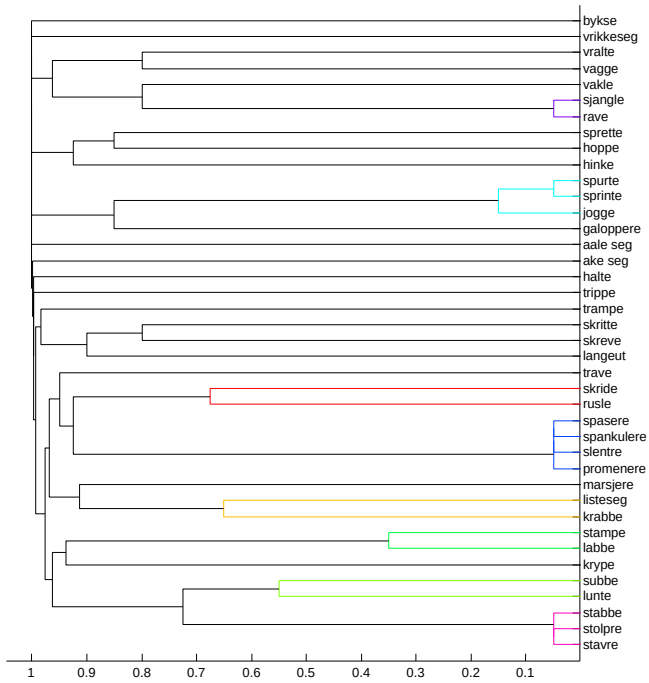


Fig. 7. Clusters based on 9 features, with Jaccard metric and Average linkage.

The clusters based on these 9 features are illustrated in Figures 7 (Jaccard-Average combination) and 8 (Euclidean-Average). Corresponding feature weights are shown in Figure 6. Figure 9 illustrates the clusters for the following 8 features: *'contact'*, *'limbs'*, *'symmetrical'*, *'sideways'*, *'stride'*, *'agent'*, *'speed'*, and *'agility'* where Euclidean metric and Average linking is used. In this experiment the feature *'effort'* has been removed, while Figure 10 illustrates the clusters when the feature *'contact'* is removed instead.

V. Discussion

The results from the computational method employed have highlighted a number of interesting features of this kind of research. Firstly, they have underscored the validity

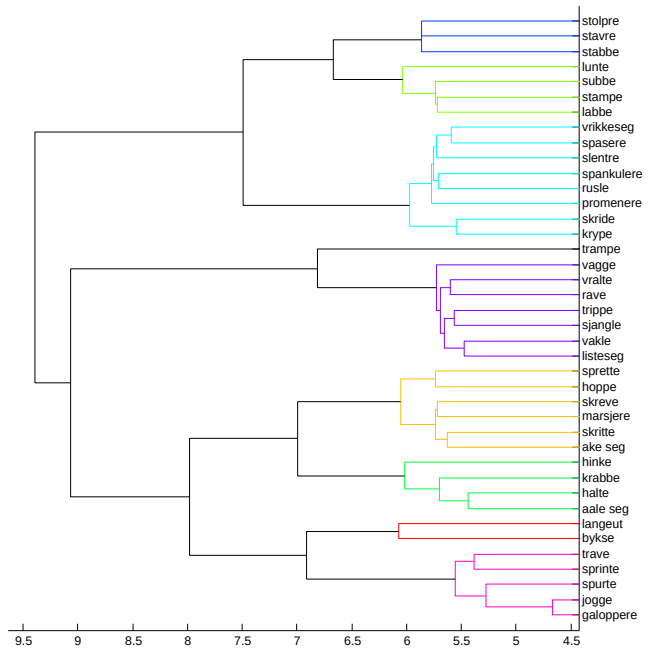


Fig. 8. Clusters based on 9 features, Euclidean metric and Average linkage was used.

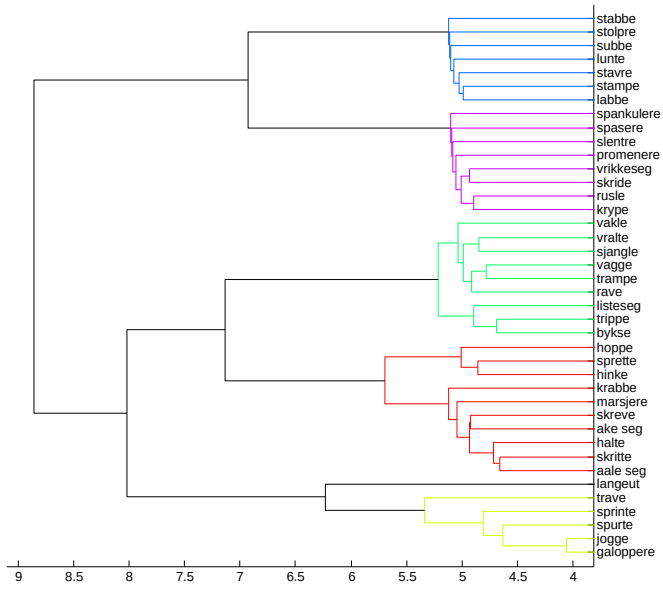


Fig. 9. Clusters based on 8 features, Euclidean metric and Average linkage is used. “Effort” is removed.

of combining human data analyses with computational methods (see also McDonald 2000). In addition, they have demonstrated that computer modelling of the data can provide useful insights for the underlying semantic similarities, as well as complement or even supplement the human analysis. Concerning the distance and linkage methods, the Euclidian Average has proven most useful in representing the underlying similarities in the data, as well as in visualising the structure of the semantic field of more specific verbs of locomotion. In contrast, the Jaccard distance metric does not seem to capture the structure of the field, and the clusters created by this method appear ad hoc and largely accidental, where one finds words describing very different kinds of motion on the same branch (e.g., vralte (move slowly swinging from side to side) and hoppe (jump)), while similar words appear on very distant nodes (e.g., the high velocity verbs). This is confirmed in our previous work as well, whereby Jaccard plots, while not particularly revealing, were good at capturing subtle details of specific similarities between isolated items.

The method of feature weighting has also proven successful and the removal of features has produced neat and succinct clusters. It is worth mentioning that feature removal has a negative side to it, since it increases the weights of certain features, while removing other features which might be relevant for an in-depth detailed analysis. Furthermore, there is a risk of capturing only the overall and more general tendencies in the structure of the semantic field at hand, while missing more subtle aspects of semantic similarity. Our tentative conclusion at this stage is that a set of 9 or 8 features is within the comfortable zone in this respect. The weighted feature cluster with 8 features is most representative of this method and reveals a graded structure of the field of locomotion, with clear-cut clusters defined on a continuum from low-speed, heavy (longer stride), non-agile motion patterns to high-velocity, agile and effort-demanding locomotions. The middle clusters reflect the importance of contact with the substrate, limb alternation, which are features carrying less weight in the 8-feature plot. This kind of graded structure has, in fact, been mentioned in the descriptions provided by the participants in the study. Some have indicated that, when arranging the groups, they have been guided by an inner structure in terms of slow effortful movement to high speed agile motion. Even though there is no exact match between the cluster obtained from the human sorting data C_{human} and the feature-weighted cluster C_{comp}, both reveal the most salient semantic features relevant for the grouping, such as speed, effort, agility, contact with the substrate. We also hypothesise, based on these results that the cluster based on the human data, reflects the individual differences and variation in what features individual speakers find most relevant for the grouping. We further hypothesise that these features are perceptual in nature and may vary according to the specific contexts in which these lexical items were acquired. For instance, for verbs that denote unsteady/swinging gaits, other factors (e.g., speed or effort) may be found irrelevant. In contrast, the cluster obtained by computer modeling and feature-weighting is based on features that the participants mentioned in the subsequent interview session and dictionary definitions of the verbs, and as such, are the result of deliberate conceptualisation. This finding is interesting in its own right and confirms usage-based accounts of language acquisition as tightly temporally and spatially-bound ([13], [14], [15], [16]).

It is worth noticing that the feature-weighted clusters based on fewer features (8 and 9) still display some anomalies. For instance, verbs like *krype* (creep), *krabbe* (crawl for human infants), *ake seg* (move butt-scooting) and *aale seg* (slither, creep like a snake) all belong in different and not immediately coherent clusters, while in the human data cluster they appear on the same branch. What these verbs share, and what is reflected in the human sorting, is the fact that all of these types of locomotion are non-default (for humans), presuppose greater contact with the substrate, in the case of aale seg, full body contact with the ground, and the use of more limbs than just the legs. We propose that the feature-weighted cluster does not reflect this similarity properly as the result of removing some of the features that underlie the similarity among the above verbs, most notably the two low-weight features: "propulsion" and "position". As observed above, this is one of the down-sides of feature removal and modeling.

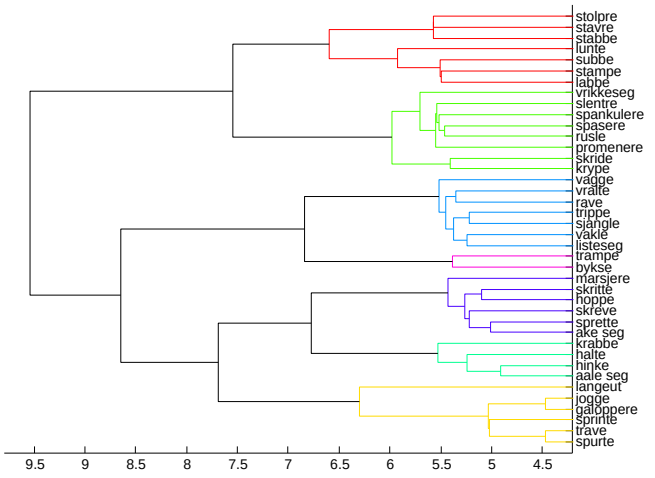


Fig. 10. Clusters based on 8 features, Euclidean metric and Average linkage is used. 'Contact' is removed.

Overall, the removal of different features from the feature-weighted plots has proven a successful step in the attempt to model similarity representations and to detect what features appear to be more important for describing similarities among verbs of motion at the specific level. In the 9 feature set, the removal of the features "contact" and "effort" has less of an influence on the clustering of verbs, and both clusters are close to the one based on the original human data. This is true, in particular, of the effect of removing "contact" from the feature set (see Figure 10). This can be taken as indication that this feature is less relevant for identifying special patterns of locomotion and the distinctions among them, and similarly for the corresponding verbs. There is a natural explanation for this fact: very few verbs in the set describe unsupported gaits (these are the hopping/jumping verbs, and in part, the running verbs),so capturing similarities or differences will not reside directly in the feature of contact. The removal of "effort" has greater, albeit inessential, consequences for the similarity plot. We observe that removing that feature has the effect of reducing the distances within the smaller clusters, while retaining the overall "bigger" similarity clusters, e.g., among the walking gait verbs, and in general, between walking and running verbs. In contrast, the removal of the feature "agility" has drastic consequences for the similarity plot and produces an altogether non-coherent clustering. An additional effect is reducing, or rather removing, the distinctions especially within the walking verbs group. We conclude that agility is an important feature in capturing similarities/differences among more specific verbs of locomotion which are below the basic level.

In conclusion, we have seen that the model is successful in identifying features relevant for the clustering of specific verbs of locomotion. In addition, we observe that the features which play a role in describing verbs of motion at the basic level, such as "contact", "speed", "effort" [5], while still relevant for the specific verbs, do not help so much in distinguishing among those verbs. It is other features, such as, most notably "agility", which are good candidates for capturing the underlying structure of the field. This result is very satisfactory, and confirms that humans resort to different sets of features in categorising the world at different levels of categorization, which differ in degree of granularity and detail (e.g., the basic level vs. the level below the basic level). This finding also aligns with recent trends in cognitive science to look at the various grain-levels of categorisation and their linguistic encoding ([17] and the papers therein).

VI. Conclusion

The results from the computational method employed have highlighted a number of interesting features of this kind of research. Firstly, they have underscored the validity of combining human data analyses with computational methods. In addition, they have demonstrated that computer modeling of the data can provide useful insights for the underlying semantic similarities, as well as complement or even supplement the human analysis. Data from applying this design to more languages is needed in order to assess fully its applications and use.

References

[1] E. S. Cordingley, "Knowledge elicitation techniques for knowledge-based systems," in *Knowledge elicitation: principle, techniques and applications*. New York, NY, USA: Springer-Verlag New York, Inc., 1989, pp. 87–175.

[2] J. Geiwitz, J. Kornell, and B. P. McCloskey, "An expert system for the selection of knowledge acquisition techniques," Santa Barbara, CA: Anacapa Sciences, 1990, technical Report 785-2.

[3] D. Roberson, I. R. L. Davies, G. G. Corbett, and M. Vandervyver, "Free-sorting of colors across cultures: Are there universal grounds for grouping?" *Journal of Cognition and Culture*, vol. 5, no. 3, pp. 349–386, 2005.

[4] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johson, and P. Boyes-Bream, "Basic objects in natural categories," *Cognitive Psychology*, vol. 8, pp. 382–439, 1976.

[5] M. Vulchanova, L. Martinez, and O. Edsberg, "A basic level category for the encoding of biological motion," in *Conceptual Spaces and the Construal of Spatial Meaning. Empirical evidence from human communication*, J. Hudson, C. Paradis, and U. Magnusson, Eds. Oxford: Oxford University Press, in press.
[6] K. Coventry, M. Vulchanova, T. Cadierno, L. Martinez, and R. Pajusalu, "Locomotion below the basic level: Sorting verbs across languages," in preparation.
[7] S. Padó and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.
[8] M. Baroni, B. Murphy, E. Barbu, and M. Poesio, "Strudel: A corpus-based semantic model based on properties and types," *Cognitive Science*, vol. 34, no. 2, pp. 222–254, 2010.
[9] S. McDonald, "Environmental determinants of lexical processing effort," PhD thesis. University of Edinburgh, 2000.
[10] E. Dabrowska, "Words as constructions," in *New Directions in Cognitive Linguistics*, E. Vyvyan and S. Pourcel, Eds. Amsterdam: John Benjamins, 2009.
[11] B. Zhang and S. Srihari, "Binary vector dissimilarity measures for handwriting identification," in *Proceedings of SPIE*, vol. 5010, 2003, pp. 28–38.
[12] S. Debnath, N. Ganguly, and P. Mitra, "Feature weighting in content based recommendation system using social network analysis," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*. New York, NY, USA: ACM, 2008, pp. 1041–1042.
[13] L. B. Smith, "Action alters shape categories," *Cognitive Science*, vol. 29, no. 4, pp. 665–679, 2005.
[14] K. Coventry and S. Garrod, "Saying, seeing and acting: The psychological semantics of spatial prepositions." Hove: Psychology Press, 2004.
[15] K. Abbot-Smith and M. Tomasello, "Exemplar-learning and schematization in a usage-based account of syntactic acquisition," *Linguistic Review*, vol. 23, no. 3, pp. 275–290, 2006.
[16] E. Dabrowska, "The mean lean grammar machine meets the human mind: Empirical investigations of the mental status of rules." in *Cognitive foundations of linguistic usage patterns. Empirical approaches*, H. Schmid and S. Handl, Eds. Berlin: Mouton de Gruyter., 2010.
[17] "Motion encoding in language," Oxford University press, in press.

Semantic Textual Entailment Recognition using UNL

Partha Pakray, Soujanya Poria, Sivaji Bandyopadhyay, and Alexander Gelbukh

***Abstract*—A two-way textual entailment (TE) recognition system that uses semantic features has been described in this paper. We have used the Universal Networking Language (UNL) to identify the semantic features. UNL has all the components of a natural language. The development of a UNL based textual entailment system that compares the UNL relations in both the text and the hypothesis has been reported. The semantic TE system has been developed using the RTE-3 test annotated set as a development set (includes 800 text-hypothesis pairs). Evaluation scores obtained on the RTE-4 test set (includes 1000 text-hypothesis pairs) show 55.89% precision and 65.40% recall for YES decisions and 66.50% precision and 55.20% recall for NO decisions and overall 60.3% precision and 60.3% recall.**

***Index Terms*—Textual Entailment, Universal Networking Language (UNL), RTE-3 Test Annotated Data, RTE-4 Test Data**

I. Introduction

RECOGNIZING Textual Entailment is one of the recent challenges of Natural Language Processing. Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing "Text" (T) and the entailed "Hypothesis" (H). T entails H if the meaning of H can be inferred from the meaning of T.

Textual Entailment has many applications in Natural Language Processing tasks: in Summarization (SUM), a summary should be entailed by the text; Paraphrases (PP) can be seen as mutual entailment between a T and a H; in Information Extraction (IE), the extracted information should also be entailed by the text; in Question Answering (QA) the answer obtained for one question after the Information Retrieval (IR) process must be entailed by the supporting snippet of text.

There were three Recognizing Textual Entailment competitions RTE-1 in 2005 [4], RTE-2 [1] in 2006 and RTE-3 [6] in 2007 which were organized by PASCAL (Pattern Analysis, Statistical Modeling and Computational Learning)—European Commission's IST-funded Network of Excellence for Multimodal Interfaces. In 2008, the fourth edition (RTE-4) of the challenge was organized by NIST (National Institute of Standards and Technology) in Text Analysis Conference (TAC). In every new competition several new features of RTE were introduced. The RTE-5 challenge in 2009 includes a separate search pilot along with the main task.

The first PASCAL Recognizing Textual Entailment Challenge (RTE-1) [4], introduced the first benchmark for the entailment recognition task. The RTE-1 dataset consists of manually collected text fragment pairs, termed text *t* (1-2 sentences) and hypothesis *h* (one sentence). The systems were required to judge for each pair whether *t* entails *h*. The pairs represented success and failure settings of inferences in various application types (termed "tasks"). In RTE-1 the various techniques used by the participating systems were word overlap, WordNet, statistical lexical relation, world knowledge, syntactic matching and logical inference.

After the success of RTE-1, the main goal of the RTE-2, held in 2006 [1], was to support the continuity of research on textual entailment. The RTE-2 data set was created with the main focus of providing more "realistic" text-hypothesis pair. As in the RTE-1, the main task was to judge whether a hypothesis H is entailed by a text. The texts in the datasets were of 1-2 sentences, while the hypotheses were one sentence long. Again, the examples were drawn to represent different levels of entailment reasoning: lexical, syntactic, morphological and logical. The main task in the RTE-2 challenge was classification—entailment judgment for each pair in the test set that represented either entailment or no entailment. The evaluation criterion for this task was accuracy—the percentage of pairs correctly judged. A secondary task was created to rank the pairs based on their entailment confidence. A perfect ranking would place all the positive pairs (for which the entailment holds) before all the negative pairs. This task was evaluated using the average precision measure [8], which is a common evaluation measure for ranking in information retrieval. In RTE-2 the techniques used by the various participating systems are Lexical Relation/ database, n-gram/ subsequence overlap, syntactic matching/ Alignment, Semantic Role labeling / FrameNet / PropBank, Logical Inference, Corpus/web-based statistics, machine learning (ML) Classification, Paraphrase and Templates, Background Knowledge and acquisition of entailment corpus.

The RTE-3 data set consisted of 1600 text-hypothesis pairs, equally divided into a development set and a test set. The same four applications from RTE-2—namely IE, IR, QA and

Manuscript received November 2, 2010. Manuscript accepted for publication January 12, 2011. This work was supported in part by the Government of India and Government of Mexico (joint DST-CONACYT project) and Government of Mexico (CONACYT 50206-H, SIP-IPN 20113295, as well as SNI and CONACYT Sabbatical program as to the fourth author).

P. Pakray, S. Poria, and S. Bandyopadhyay are with the Computer Science and Engineering Department, Jadavpur University, Kolkata, India (e-mail: parthapakray@gmail.com, soujanya.poria@gmail.com, sbandyopadhyay@cse.jdvu.ac.in).

A. Gelbukh is with the Faculty of Law, Waseda University, Tokyo, Japan, on Sabbatical leave from the Center for Computing Research, National Polytechnic Institute, Mexico City, Mexico (e-mail: gelbukh@gelbukh.com).

SUM—were considered as settings or contexts for the pair's generation. 200 pairs were selected for each application in each data set. Each pair was annotated with its related task (IE/IR/QA/SUM) and entailment judgment (YES/NO). In addition, an optional pilot task, called "Extending the Evaluation of Inferences from Texts" was set up by the NIST, in order to explore two other sub-tasks closely related to textual entailment: differentiating unknown entailment from identified contradictions and providing justifications for system decisions. In the first sub-task, the idea was to drive systems to make more precise informational distinctions, taking a three-way decision between "YES", "NO" and "UNKNOWN", so that a hypothesis being unknown on the basis of a text would be distinguished from a hypothesis being shown false/contradicted by a text.

In RTE-4 [5], no development set was provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications—namely IE, IR, QA and SUM—were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. A major difference with respect to RTE-3 was that the RTE-4 data set consisted of 1000 T-H pairs, instead of 800. In RTE-4, the challenges were classified as two-way task and three-way task. The two-way RTE task was to decide whether:

1) T entails H—in which case the pair will be marked as ENTAILMENT;
2) T does not entail H—in which case the pair will be marked as NO ENTAILMENT.

The three-way RTE task was to decide whether:

3) T entails H—in which case the pair was marked as ENTAILMENT,
4) T contradicts H—in which case the pair was marked as CONTRADICTION,
5) The truth of H could not be determined on the basis of T—in which case the pair was marked as UNKNOWN.

In every new competition several new features of RTE were introduced. The TAC RTE-5 [2] challenge in 2009 includes a separate search pilot along with the main task. The TAC RTE-6 challenge[1], in 2010, includes the Main Task and Novelty Detection Task along with RTE-6 KBP Validation Pilot Task. The RTE-6 does not include the traditional RTE Main Task, which was carried out in the first five RTE challenges, i.e., there was no task to make entailment judgments over isolated T-H pairs drawn from multiple applications. In 2010, Parser Training and Evaluation using Textual Entailment [9] was organized by SemEval-2. We have developed our own RTE system and have participated in TAC RTE-5 and Parser Training and Evaluation using Textual Entailment as part of SemEval-2 and also in TAC RTE-6.

In the present paper, a 2-way semantic textual entailment recognition system has been described that has been trained on the 2-way RTE-3 test gold set and then tested on the RTE-4 test set. UNL Expressions are described in Section 2. Section 3 describes semantic based RTE system architecture. The experiment carried out on the development and test data sets are described in Section 4 along with the results. The conclusions are drawn in Section 5.

II. UNL Expressions

Universal Networking Language (UNL) is an artificial language that can be used as a pivot language in machine translation systems or as a knowledge representation language in information retrieval applications. The UNL [3, 7] expresses information or knowledge in the form of semantic network with hyper-node. UNL semantic network is made up of a set of binary relations, each binary relation is composed of a relation and two Universal Words (UWs) that hold the relation. A binary relation of UNL is expressed in the format shown in Table I.

TABLE I
UNL Relation

<relation> (<uw1>, <uw2>)

In <relation>, one of the relations defined in the UNL Specifications is described. In <uw1> and <uw2>, the two UWs that hold the relation given at <relation> are described.

All binary relations that compose a UNL expression have directions, and the semantic network of a UNL expression is a directed hyper-graph.

A. *UNL expression hyper-graph*

Each UNL expression is a semantic hyper-network. That is, each node of the graph, <uw1> and <uw2> of a binary relation, can be replaced with a semantic network. Such a node consists of a semantic network of a UNL expression and is called a "scope". A scope can be connected with other UWs or scopes. Each UNL expression in a scope is distinguished from others by assigning an ID to the <relations> of the set of binary relations that belong to the scope.

The general description format of binary relations for a hyper-node of UNL expression is in Table II, where:

– <scope-id> is the ID for distinguishing a scope. <scope-id> is not necessary to be specified when a binary relation does not belong to any scope.
– <node1> and <node2> can be a UW or a <scope node>.
– A <scope node> is given in the format ": <scope-id>".

TABLE II
UNL Expression

<relation>:<scope-id> (<node1>, <node2>)

An example UNL expression for hypothesis is given in Table III.

The EnConverter and DeConverter are the core software in the UNL System. The EnConverter converts natural language sentences into UNL Expressions. The DeConverter converts

1 http://www.nist.gov/tac/2010/RTE/index.html

TABLE III
UNL RELATION

```
{org:en}
UN peacekeepers abuse children.
{/org}
{unl}
mod(peacekeeper(icl>defender>thing).@pl,un(icl>world_organization>
    thing,equ>united_nations))
agt(abuse(icl>treat>do,equ>mistreat,agt>person,obj>living_thing).@entry.
    @present,peacekeeper(icl>defender>thing).@pl)
obj(abuse(icl>treat>do,equ>mistreat,agt>person,obj>living_thing).@entry.
    @present,child(icl>juvenile>thing).@pl)
{/unl}
```

UNL Expressions to natural language sentences. Both the EnConverter and DeConverter perform their functions based on a set of grammar rules and a word dictionary of a target language.

B. *UNL Relations*

Some of the UNL Relations are shown in Table IV. We used the Expanded Rules in Table VIII. These expanded rules, based on the present UNL Expression, have been developed from the RTE-3 test annotated corpus. Then these rules are applied on RTE-4 test set. Currently the system has 35 expanded rules.

TABLE IV
UNL RELATION DESCRIPTION

Relations Name	Details
agt (agent)	defines a thing that initiates an action.
mod (modification)	defines a thing that restricts a focused thing.
nam (name)	defines a name of a thing.
plc (place)	defines a place where an event occurs, or a state that is true, or a thing that exists.
plt (final place)	defines a place where an event ends or a state that is false.
tim (time)	defines the time an event occurs or a state that is true.
tmf (initial time)	defines the time an event starts or a state that is true.
tmt (final time)	defines a time an event ends or a state that is false.
to (destination)	defines a final state of a thing or a final thing (destination) associated with the focused thing.
src (source: initial state)	defines the initial state of an object or thing initially associated with the object of an event.
obj(affected thing))	defines a thing in focus that is directly affected by an event or state.

III. SYSTEM DESCRIPTION

In this section, we describe our semantic based textual entailment system. The system accepts pairs of text snippets (text and hypothesis) at the input and gives a value at the output: YES if the text entails the hypothesis and NO otherwise. The architecture of the proposed system is described in Fig. 1.

A. *Pre-processing*

The system accepts pairs of text snippets (text and hypothesis) at the input and gives the output: YES if the text entails the hypothesis and NO otherwise. An example text-hypothesis pair from the RTE-3 test annotated set which is used as a development set is shown in Table V.

TABLE V
RTE-3 TEST ANNOTATED SET

```
<pair id="12" entailment="YES" task="IE" length="short" >
<t>Judge Drew served as Justice until Kennon returned to claim his seat in
    1945.</t>
<h>Kennon served as Justice.</h>
</pair>
```

In the development set, the following expressions were replaced: “aren’t” with “are not”, “didn’t” with “did not”, “doesn’t” with “does not”, “won’t” with “will not”, “don’t” with “do not”, “hasn’t” with “has not”, “isn’t” with “is not”, “couldn’t” with “could not”, “ă” with “a”, "á" with “a”, "š" with “s”, "ž" with “z” and "ó" with “o”. These expressions are either abbreviations or include special characters for which the dependency parser gives erroneous results. It has also been observed that escape characters like ", …, ‘ and & are present in the text and the hypothesis parts and these were removed. All the above pre-processing methods were also applied on the RTE-4 test set.

B. *UNL Enconverter Module*

In this module, we convert the text and hypothesis pair into UNL expressions[2]. For example, the UNL expression for the hypothesis in Table V is shown in Table VI, and the UNL Graph for this hypothesis is shown in Fig. 2.

TABLE VI
UNL EXPRESSION FOR RTE-3 TEST ANNOTATED SET HYPOTHESIS

```
[S:00]
{org:en}
Kennon served as Justice
{/org}
{unl}
aoj(serve(icl>be,obj>uw,aoj>thing,ben>thing).@entry.@past,kennon)
obj(serve(icl>be,obj>uw,aoj>thing,ben>thing).@entry.@past,justice
    (icl>righteousness>thing,ant>injustice).@maiuscul)
{/unl}
[/S]
```

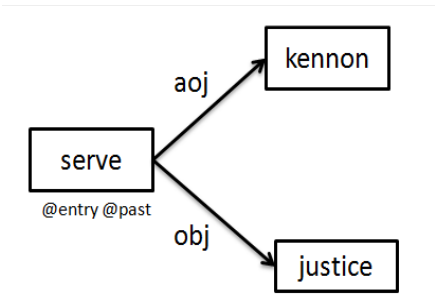


Fig. 2. UNL Hyper-graph.

In this case the output is filtered to retain the UNL relations (semantic relations) only which is shown in Table VII.

TABLE VII
UNL EXPRESSION FOR RTE-3 TEST ANNOTATED SET HYPOTHESIS

```
aoj(serve, kennon)
obj(serve, justice)
```

2 http://unl.ru/deco.html

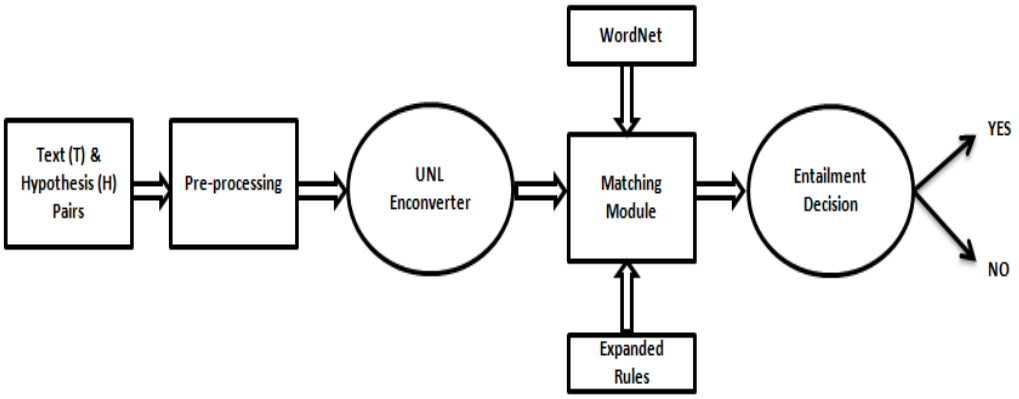


Fig. 1. Semantic Textual Entailment System.

C. Matching Module

After UNL relations are identified for both the text and the hypothesis in each pair, the hypothesis UNL relations are compared with the text UNL relations. The different features that are compared are explained below. In all comparisons, a matching score of 1 is considered when the complete UNL relation along with all of its arguments matches in both the text and the hypothesis. In case of a partial match for a UNL relation, a matching score of 0.5 is assumed. We used the partial match in Rule 3 only.

TABLE VIII
UNL EXPRESSION

Previous Relation	Expand Relation	Example
mod(x,y)	aoj(y,x)	*Red Leaf ⇒ Leaf is Red*
pos(x,y)	mod(y,x)	*Newton's Law ⇒ Newton Law*
aoj(x,y), aoj(y,z)	aoj(x,z)	*He is a boy. A boy is a man. ⇒ He is a man.*
pos(x,y), agt(z,x)	agt(z,y)	*Chief Minister of West Bengal said the thing. ⇒ West Bengal said the thing.*
man(x,y),bas(x,y)	aoj(x,z)	*A rose is more beautiful than tulip. ⇒ Rose is beautiful.*
ins(x,y)	ins(x,z), if z is a hypernym of y.	*He sang with a guitar. ⇒ He sang with an instrument.*
pos(x,y)	iof(x,y)	*Tokyo is a city in Japan. ⇒ Tokyo is a city of Japan.*
and(x,y),and(y,z)	and(y,z)	*You and me., Me and Ramesh. ⇒ Ramesh and you.*

Rule 1: Match Relation = (Number of hypothesis UNL relations that match with text / Number of hypothesis UNL relations)

If Match Relation is above 60%, then this pair is marked as "YES", otherwise as "NO".

Rule 2: If the above Match Relation entailment value is "NO" then we apply the expanded rule given below in both the hypothesis and the text file.

Match Relation (Expand rule) = (Number of hypothesis UNL relations that match with text (obtained from Rule 1) + Number of hypothesis UNL relations that match with text by Expand rule / Number of hypothesis UNL relations).

Expand rules are applicable to those UNL relations that do not match in Rule 1. If Match Relation (Expand rule) is above 60%, then this pair is marked as "YES", otherwise as "NO".

Rule 3: If Match Relation (Expand rule) entailment value is "NO" then we apply the Rule 3 as given below in both the hypothesis and the text file.

Match Relation (Partial Expand rule) = (Number of hypothesis UNL relations that match with text (obtained from Rule 1) + Number of hypothesis UNL relations that match with text by Expand rule (obtained from Rule 2) + Number of hypothesis UNL relation match with text by WordNet synonym / Number of hypothesis UNL relations).

If Match Relation (Partial Expand rule) is above 60% then this pair marked as "YES", otherwise as "NO".

IV. EXPERIMENTAL RESULTS

In RTE-4, no development set was provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the systems. Four applications—namely IE, IR, QA and SUM—were considered as settings or contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. The RTE-3 test annotated set was used to train our entailment system to identify the threshold values for the various measures towards entailment decision. The two-way RTE-3 test annotated set consisted of 800 text–hypothesis pairs. The RTE-4 test set consisted of 1000 text–hypothesis pairs.

Two baseline systems have been developed in the present task. The Baseline-1 system assigns YES tag to all the text-hypothesis pairs and the Baseline-2 system assigns NO tag to all the text hypothesis pairs.

TABLE IX
BASELINE SYSTEMS FOR RTE-3 DEVELOPMENT SET AND RTE-4 TEST SET: # STANDS FOR THE NUMBER OF DECISIONS, P FOR PRECISION

	Decision	Gold standard	Baseline-1		Baseline-2	
			#	P, %	#	P, %
RTE-3	YES	410	800	51.25	0	0
Development Set	NO	390	0	0	800	48.75
RTE-4	YES	500	1000	50.00	0	0
Test Set	NO	500	0	0	1000	50.00

The results obtained on Baseline-1 and Baseline-2 systems on the RTE-3 development data set and the RTE-4 test data set are shown in Table IX.

In our textual entailment system, the method was run separately on the RTE-3 test annotated set and two-way entailment ("YES" or "NO") decisions were obtained for each text-hypothesis pair. Experiments were carried out to measure the performance of the final RTE system. It is observed that the precision and recall measures of the final RTE system are best when final entailment decision is based on entailment value (YES/NO) results with threshold value 0.60. The results on the RTE-3 test annotated data set are shown in Table X.

TABLE X
UNL RTE-3 DEVELOPMENT SET STATISTICS FOR OUR SYSTEM WITH DIFFERENT THRESHOLD VALUES

		Threshold		
		0.50	0.60	0.70
"YES"	System	572	481	461
	System ∩ Gold	313	278	257
	Gold	410	410	410
	Precision, %	54.72	57.79	55.74
	Recall, %	76.34	67.80	62.68
"NO"	System	228	319	339
	System ∩ Gold	131	204	186
	Gold	390	390	390
	Precision, %	57.45	63.94	54.86
	Recall, %	33.58	52.30	47.69

Experiments were carried out to measure the performance of the final RTE system. The results on the RTE-3 test annotated set for "YES" and "NO" entailment decisions are shown in Table XI.

TABLE XI
RTE-3 TEST ANNOTATED DATA SET STATISTICS FOR OUR SYSTEM, WITH THRESHOLD VALUE 0.60

Entailment Decision	Gold standard	System, correct	System, total	Precision	Recall
YES	410	278	481	57.79%	67.80%
NO	390	204	319	63.94%	52.30%
Total	800	482	800	60.25%	60.25%

The results on RTE-4 test set are shown in Table XII.

TABLE XII
RTE-4 TEST SET STATISTICS FOR OUR SYSTEM, WITH THRESHOLD VALUE 0.60

Entailment Decision	Gold standard	System, correct	System, total	Precision	Recall
YES	500	327	585	55.89%	65.40%
NO	500	276	415	66.50%	55.20%
OVERALL	1000	603	1000	60.30%	60.30%

V. CONCLUSION

Our results show that a Semantic-based approach appropriately tackles the textual entailment problem. Experiments have been initiated for a semantic and syntactic based RTE task.

The next step is to carry out detailed error analysis of the present system and identify ways to overcome the errors. In the present task, the final RTE system has been optimized for the entailment YES/NO decision using the development set. The role of the application setting for the RTE task has also not yet been looked into. This needs to be experimented in the future. The two-way task has to be upgraded to the three-way task.

Finally, given that graph-matching is a computationally expensive task [10], we plan to improve the computational efficiency of our algorithm.

REFERENCES

[1] R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor, "The Second PASCAL Recognising Textual Entailment Challenge," in *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy. 2006.

[2] L. Bentivogli, I. Dagan, H.T. Dang, D. Giampiccolo, and B. Magnini, "The Fifth PASCAL Recognizing Textual Entailment Challenge," in *TAC 2009 Workshop*, National Institute of Standards and Technology Gaithersburg, Maryland USA. 2009.

[3] J. Cardeñosa, A. Gelbukh, E. Tovar (eds.), *Universal Networking Language: Advances in Theory and Applications*, IPN. 2005; www.gelbukh.com/UNL-book.

[4] I. Dagan, O. Glickman, and B. Magnini, "The PASCAL Recognising Textual Entailment Challenge," in *Proceedings of the PASCAL RTE Challenge*. 2005.

[5] D. Giampiccolo, H. T. Dang, B. Magnini, I. Dagan, and E. Cabrio, "The Fourth PASCAL Recognizing Textual Entailment Challenge," in *TAC 2008 Proceedings*. 2008.

[6] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan, "The Third PASCAL Recognizing Textual Entailment Challenge," in *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic. 2007.

[7] UNDL Foundation, *Universal networking language (UNL) specifications*, edition 2006, August 2006. http://www.undl.org/unlsys/unl/unl2005-e2006/.

[8] E.M. Voorhees and D. Harman, "Overview of the Seventh Text Retrieval Conference," in *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*. NIST Special Publication. 1999.

[9] D. Yuret, A. Han, and Z. Turgut, "SemEval-2010 Task 12: Parser Evaluation using Textual Entailments," in *Proceedings of the SemEval-2010 Evaluation Exercises on Semantic Evaluation*. 2010.

[10] G. Zhao, M. Petridis, G. Sidorov, and J. Ma, "A critical examination of node similarity graph matching algorithm," *Research in computing science*, vol. 40, pp. 73–82, 2008.

Examining the Validity of Cross-Lingual Word Sense Disambiguation

Els Lefever and Veronique Hoste

***Abstract*—This paper describes a set of experiments in which the viability of a classification-based Word Sense Disambiguation system that uses evidence from multiple languages is investigated. Instead of using a predefined monolingual sense-inventory such as WordNet, we use a language-independent framework and start from a manually constructed gold standard in which the word senses are made up by the translations that result from word alignments on a parallel corpus. To train and test the classifier, we used English as an input language and we incorporated the translations of our target words in five languages (viz. Spanish, Italian, French, Dutch and German) as features in the feature vectors. Our results show that the multilingual approach outperforms the classification experiments where no additional evidence from other languages is used. These results confirm our initial hypothesis that each language adds evidence to further refine the senses of a given word. This allows us to develop a proof of concept for a multilingual approach to Word Sense Disambiguation.**

***Index Terms*—Word Sense Disambiguation, multilingual, cross-lingual.**

I. Introduction

WORD Sense Disambiguation (WSD) is the NLP task that consists in selecting the correct sense of a polysemous word in a given context. For a detailed overview of the main WSD approaches we refer to Agirre and Edmonds [1] and Navigli [2]. State-of-the-art WSD systems are mainly supervised systems, trained on large sense-tagged corpora, where human annotators have labeled each instance of the target word with a label from a predefined sense inventory such as WordNet [3]. Two important problems arise with this approach. Firstly, large sense-tagged corpora and sense inventories are very time-consuming and expensive to build. As a result they are extremely scarce for languages other than English. In addition, there is a growing conviction within the WSD community that WSD should not be tested as a stand-alone NLP task, but should be integrated in real applications such as Machine Translation and cross-lingual information retrieval [4].

In this paper, we describe the construction of a multilingual WSD system that takes an English ambiguous word and its context as input, and outputs correct translations for this ambiguous word in a given focus language. For our experiments we trained a classifier for five focus languages (viz. Italian, German, Dutch, Spanish and French). In addition to a set of local context features, we included the translations in the four other languages (depending on the focus language of the classifier) in the feature vector. All translations are retrieved from the parallel corpus Europarl [5].

Using a parallel corpus, such as for example Europarl, instead of human defined sense-labels offers some advantages: (1) for most languages we do not have large sense-annotated corpora or sense inventories, (2) using corpus translations should make it easier to integrate the WSD module into real multilingual applications and (3) this approach implicitly deals with the granularity problem, as fine sense distinctions (that are often listed in electronic sense inventories) are only relevant in case they get lexicalized in the target translations. The idea to use translations from parallel corpora to distinguish between word senses is based on the hypothesis that different meanings of a polysemous word are lexicalized across languages [6], [7]. Many WSD studies have already shown the validity of this cross-lingual evidence idea. Most of these studies have focused on bilingual WSD (E.g.[8], [9], [10]) or on the combination of existing WordNets with multilingual evidence (E.g. [11]).

In order to use the parallel texts to train a WSD classifier, most systems lump different senses of the ambiguous target word together if they are translated in the same way (E.g. Chan and Ng [12]), which reflects the problem of assigning unique translations to each sense of a noun. If we take for instance the English word *mouse*, this is translated in French as *souris*, both for the animal and the computer sense of the word. In order to construct and refine a multilingual sense inventory reflecting the different senses of a given word, more translations are required to increase the chance that the different word senses are lexicalized differently across the different languages. To our knowledge, however, it has not been shown experimentally if and how much multilingual evidence from a parallel corpus indeed helps to perform classification-based WSD for a given target language. In the experiments reported in this paper, we included evidence from up to 4 languages into the feature vectors of a multilingual lexical sample WSD classifier.

The remainder of this paper is organized as follows. Section II describes the data set we used for the experiments. Section III presents the construction of the feature vectors, and gives more insights in the classifier that was built. Section IV gives an overview of the experiments and we finally draw conclusions and present some future research in Section V.

Manuscript received November 6, 2010. Manuscript accepted for publication January 12, 2011.

The authors are with the LT3, University College Ghent, Groot-Brittanniëlaan 45, Ghent, Belgium and Dpt. of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281(S9), Ghent, Belgium (e-mail: {Els.Lefever,Veronique.Hoste}@hogent.be).

II. Data

In order to construct our sense inventory, we extracted the translations of our ambiguous target words from the parallel corpus Europarl [5]. We selected 6 languages from the 11 European languages represented in the corpus, viz. English (our target language), Dutch, French, German, Italian and Spanish. As our approach is both language- and corpus-independent, and all steps can be run in an automatic way, we can easily add other languages and extend or replace the corpus that was used.

All Europarl data were already sentence-aligned using a tool based on the Gale and Church algorithm [13], which was part of the corpus. We only considered the intersected 1-1 sentence alignments between English and the five other languages (see also [11] for a similar strategy). The experiments were performed on a lexical sample of five ambiguous words, being *bank*, *plant*, *movement*, *occupation* and *passage*, which were collected in the framework of the SemEval-2 Cross-Lingual Word Sense Disambiguation task. The six-language sentence aligned corpus, as well as the test set and corresponding gold standard, can be downloaded from the task website[1].

After the selection of all English sentences containing these target nouns and the aligned sentences in the five target languages, we used GIZA++ [14] word alignment on the selected sentences to retrieve the set of possible translations for our ambiguous target words. All alignments were manually checked afterwards. In cases where one single target word (E.g. **occupation**) led to a multiword translation (e.g *actividad profesional* in Spanish) or to a compound (e.g *beroepsbezigheden* in Dutch and *Berufstätigkeit* in German), we kept the multi-part translation as a valid translation suggestion.

All sentences containing the target words were preprocessed by means of a memory-based shallow parser (MBSP) [15], that performs tokenization, Part-of-Speech tagging and text chunking. On the basis of these preprocessed data, we built a feature vector which contains information related to the target word itself as well as local patterns around the target word. Table I shows the size of the instance base for each of the ambiguous words, whereas Figure 1 lists the number of classes per ambiguous target word in the five focus languages.

TABLE I
SIZE OF THE INSTANCE BASE PER AMBIGUOUS TARGET WORD

	Number of instances
bank	4029
movement	4222
occupation	634
passage	238
plant	1631

Figure 1 also suggests that due to the high number of unique translations in Dutch and German, mainly due to their compounding strategies, the classification task will be especially challenging for these two languages.

As Figure 1 shows, the polysemy of the target words is considerably high in all five target languages. Even for the romance languages, where the number of compound translations is rather low, the classifier has to choose from a substantial number of possible classes. Example 1 illustrates this by listing the French translations that were retrieved for the English word *plant* (*NULL* refers to a null link from the word alignment):

(1) centrale, installation, plante, usine, végétal, NULL, phytosanitaire, entreprise, incinérateur, station, pesticide, site, flore, unité, atelier, plant, phytopharmaceutique, établissement, culture, réacteur, protéagineux, centre, implantation, oléoprotéagineux, équipement, horticulture, phytogénétique, exploitation, végétation, outil, plantation, sucrerie, société, fabrique, four, immobilisation, céréale, espèce, séchoir, production, claque, arsenal, ceps, poêle, récolte, plateforme, artémisinine, fabrication, phytogénéticien, oléagineux, glacière, espèce végétale, chou, tranche, Plante, installation incinérateur.

III. Experimental Set-up

We consider the WSD task as a classification task: given a feature vector containing the ambiguous word and the context as features, a classifier predicts the correct sense (or translation in our case) for this specific instance.

A. Feature Vectors

For our initial feature set we started off with the traditional features that have shown to be useful for WSD [1]:

- features related to the **target word itself** being the word form of the target word, the lemma, Part-of-Speech and chunk information
- **local context features** related to a window of three words preceding and following the target word containing for each of these words their full form, lemma, Part-of-Speech and syntactic dependencies.

In addition to these well known WSD features, we integrated the translations of the target word in the other languages (Spanish, German, Italian, Dutch and French depending on the desired classification output) as separate features into the feature vector. Example 2 lists the feature vector for one of the instances in the training base of the Dutch classifier. The first features contain the word form, PoS-tag and chunk information for the three words preceding the target word, the target word itself and for the three words following the target word. In addition we added the aligned translations for the target word in the four additional languages (being German, Spanish, Italian and French for the Dutch classifier). The last field contains the classification label, which is the aligned Dutch translation in this case.

[1] http://lt3.hogent.be/semeval/

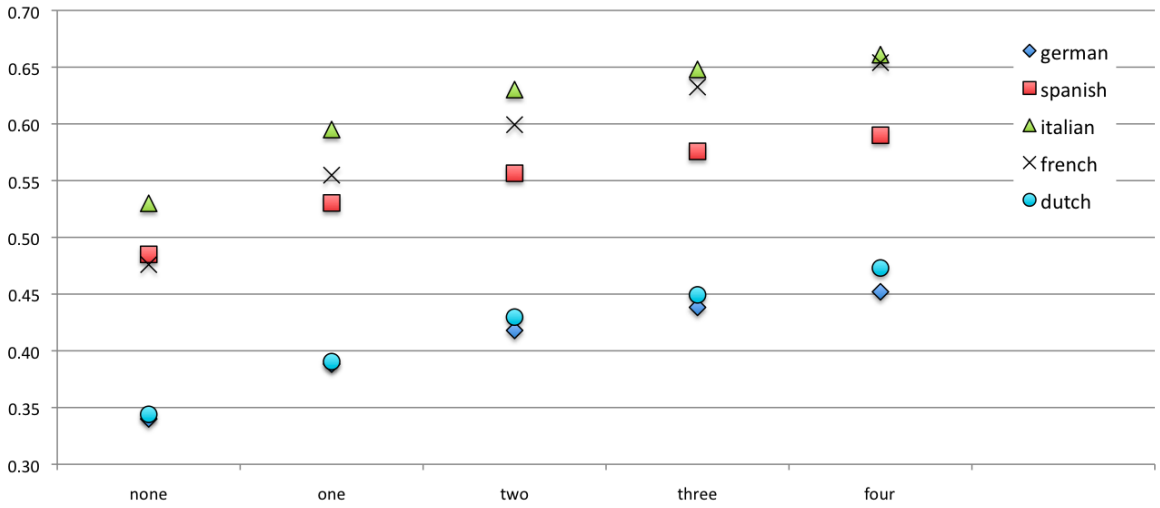


Fig. 1. Number of unique translations per language and per ambiguous word.

(2) English input sentence for the word **bank**:
This is why the Commission resolved on raising a complaint against these two banks at its last meeting, and I hope that Parliament approves this step.
Feature vector:
against these two against these two IN DT CD I-PP I-NP I-NP banks bank NNS I-NP at its last at its last IN PRP JJ I-PP I-NP I-NP Bank banco banca banque bank

Incorporating the translations in our feature vector allows us to develop a proof of concept for a multilingual approach to Word Sense Disambiguation. This multilingual approach will consist of two steps: (1) we first examine whether evidence from different languages can lead to better sense discrimination (which is the scope of this paper) and (2) in a following step we will then introduce additional cross-lingual evidence (bag-of-words features containing all content words from the aligned translations) in the feature vectors for our WSD classifier. An automatic sense discrimination step can then be applied on the training feature base.

Unsupervised approaches to sense discrimination know a long research history. The idea to use distributional methods to cluster words that appear in similar contexts corpora has been succesfully applied on monolingual corpora (E.g. [16], [17]), as well as on parallel corpora. Previous research on parallel corpora [18], [7] confirmed the use of cross-lingual lexicalization as a criterion for performing sense discrimination. Whereas in previous research on cross-lingual WSD the evidence from the aligned sentences was mainly used to enrich WordNet information, our approach does not require any external resources. With our experiments we want to examine to which extent evidence from other languages, without additional information from external lexical resources, helps to detect correct sense distinctions that result in a better WSD classification output (or translation in our case).

B. Classification

To train our WSD classifier, we used the memory-based learning (MBL) algorithms implemented in TIMBL [19], which have been shown to perform well on WSD [20]. We performed heuristic experiments to define the parameter settings for the classifier, leading to the selection of the Jeffrey Divergence distance metric, Gain Ratio [21] feature weighting and k = 7 as number of nearest distances. In future work, we plan to use a genetic algorithm to perform joint feature selection and parameter optimization per ambiguous word [22].

IV. Evaluation

For the evaluation, we performed 10-fold cross-validation on the instance bases. As a baseline, we selected the most frequent translation that was given by the automatic word alignment. We added the translations in the other languages that resulted from the word alignment as features to our feature vector and built classifiers for each target word for all five supported languages. Since we aim to investigate the impact of cross-lingual evidence on WSD, we deliberately chose to use the manually verified gold standard word alignments. Our classification results can thus be considered as an upper bound for this task, as the automatic word alignments will presumably lead to lower performance figures.

An overview of the classification results for the romance languages (French, Italian, Spanish) can be found in Table II, whereas the classification results for Dutch and German are in Table III. Figure 2 illustrates the classification results per language for 2 ambiguous words, viz “bank” and “plant” when averaging over the translations in the feature vector.

The results show that even the simple classifier which does not incorporate translation features, beats the most frequent translation baseline for all languages (except for *occupation* in Spanish and Italian), although we can improve a lot on the

TABLE II

FRENCH (TOP LEFT), ITALIAN (TOP RIGHT) AND SPANISH (BOTTOM LEFT) RESULTS FOR A VARYING NUMBER OF TRANSLATION FEATURES INCLUDING THE OTHER FOUR LANGUAGES VIZ. ITALIAN (I), SPANISH (E), GERMAN (D), DUTCH (N) AND FRENCH (F)

French					
	bank	move-ment	occu-pation	passage	plant
Baseline	55.8	44.7	75.5	50.0	20.7
all four translation features					
IEDN	**84.9**	**71.7**	**82.8**	**60.3**	**65.4**
Three translation features					
I,E,D	84.5	70.9	80.8	59.5	63.7
E,D,N	84.0	70.7	81.6	59.1	63.7
I,D,N	83.9	70.7	82.0	59.1	61.3
I,E,N	84.6	71.3	81.2	57.4	64.3
Two translation features					
E, D	83.2	69.2	80.0	59.9	60.8
I, D	83.1	69.8	80.1	58.7	58.8
D, N	82.8	69.1	80.9	57.4	58.6
I, E	84.3	69.8	80.0	57.8	61.0
E, N	83.2	69.8	80.5	57.4	61.0
I, N	83.2	70.1	81.1	57.8	59.4
One translation feature					
D	81.4	67.5	78.9	58.7	54.0
E	83.0	67.7	79.2	56.5	56.4
I	82.4	68.4	79.5	57.4	56.1
N	82.0	68.0	80.5	57.4	55.4
No translation features					
none	83.5	65.6	76.5	55.3	47.6
Only translation features					
only	85.8	73.3	82.8	62.9	69.0

Italian					
	bank	move-ment	occu-pation	passage	plant
Baseline	54.6	51.9	78.7	37.1	32.8
all four translation features					
EFDN	**83.1**	**80.2**	**81.1**	**40.1**	**66.1**
Three translation features					
E,F,D	82.7	79.6	81.1	40.1	65.1
F,D,N	82.8	79.7	79.2	40.9	64.2
E,D,N	82.6	79.2	81.0	40.5	64.6
E,F,N	82.8	80.0	81.0	40.5	65.3
Two translation features					
F, D	82.0	78.6	79.3	40.5	63.4
E, D	81.8	78.5	80.9	40.5	62.1
D, N	81.4	77.8	78.5	40.9	62.4
E, F	82.3	79.5	80.9	40.1	64.3
F, N	82.4	79.0	79.2	41.4	63.2
E, N	82.1	78.7	80.1	40.1	62.7
One translation feature					
D	80.0	76.8	77.9	40.5	59.4
F	81.4	78.0	79.2	40.9	61.1
E	81.4	77.5	80.6	38.4	58.1
N	80.9	77.2	78.1	39.7	59.4
No translation features					
none	79.5	75.2	78.1	38.0	53.0
Only translation features					
only	83.9	81.4	81.6	42.6	67.3

Spanish					
	bank	move-ment	occu-pation	passage	plant
Baseline	58.8	51.0	81.6	24.1	30.1
all four translation features					
IFDN	**90.0**	**80.8**	**83.0**	**38.0**	**59.0**
Three translation features					
I,F,D	89.6	80.6	82.8	35.9	58.6
F,D,N	89.1	79.6	82.7	37.6	57.1
I,D,N	89.4	79.4	82.4	37.6	55.9
I,F,N	89.8	80.3	82.7	35.4	58.7
Two translation features					
F, D	88.9	79.1	82.7	35.9	55.9
I, D	88.7	79.0	82.4	36.3	54.3
D, N	88.0	78.0	82.0	38.0	53.7
I, F	89.4	79.9	82.5	34.2	57.8
F, N	89.0	79.2	82.2	35.4	57.3
I, N	89.3	78.6	82.4	34.2	54.9
One translation feature					
D	87.2	77.3	82.2	37.1	50.8
F	88.7	78.3	82.7	34.2	55.1
I	88.7	78.3	81.6	32.5	53.6
N	87.7	77.1	81.9	34.6	52.6
No translation features					
none	86.5	75.8	80.6	32.9	48.5
Only translation features					
only	89.9	82.0	83.0	40.9	63.4

TABLE III

DUTCH (LEFT) AND GERMAN (RIGHT) RESULTS FOR A VARYING NUMBER OF TRANSLATION FEATURES INCLUDING THE OTHER FOUR LANGUAGES VIZ. ITALIAN (I), SPANISH (E), GERMAN (D), DUTCH (N) AND FRENCH (F)

Dutch					
	bank	move-ment	occu-pation	passage	plant
Baseline	33.4	46.7	60.6	26.7	12.0
all four translation features					
IEDF	**80.3**	**65.8**	**69.3**	**36.3**	**47.3**
Three translation features					
I,E,D	80.0	65.1	68.9	35.0	44.2
E,D,F	79.4	65.2	69.0	34.6	45.8
I,D,F	79.4	65.5	69.2	36.3	45.2
I,E,F	79.1	63.7	68.2	35.4	44.5
Two translation features					
E, D	79.2	64.4	67.6	35.0	45.2
I, D	79.0	64.3	68.5	34.6	42.7
D, F	78.8	64.9	68.8	35.0	43.8
I, E	79.0	62.9	66.3	34.6	41.2
E, F	78.4	63.3	67.7	34.6	42.7
I, F	78.0	63.1	68.2	35.0	42.2
One translation feature					
D	77.8	63.5	67.6	35.0	40.4
E	78.1	62.1	65.3	33.3	37.1
I	77.7	62.1	66.3	33.8	38.9
F	77.3	62.1	67.6	33.8	39.8
No translation features					
none	76.6	60.8	65.2	31.7	34.4
Only translation features					
only	80.0	64.1	69.6	34.6	47.3

German					
	bank	move-ment	occu-pation	passage	plant
Baseline	36.7	32.3	39.0	20.3	14.0
all four translation features					
IEFN	**82.8**	**57.1**	**48.3**	**32.9**	**45.2**
Three translation features					
I,E,N	82.5	57.0	47.9	31.2	44.0
E,F,N	82.5	57.2	47.7	32.1	43.9
I,E,F	81.7	55.8	47.5	31.6	42.9
F,I,N	82.6	57.2	48.3	31.6	44.5
Two translation features					
E, F	81.6	55.6	45.5	31.2	41.1
I, F	81.6	55.5	46.9	31.2	41.6
F, N	82.3	56.9	47.2	30.4	42.9
I, E	81.6	55.3	46.4	29.5	41.1
E, N	82.2	56.6	46.7	30.0	41.6
I, N	82.2	57.1	48.0	30.0	42.5
One translation feature					
F	81.1	54.8	45.5	30.0	39.2
E	81.1	54.7	43.6	28.7	36.6
I	81.3	55.1	45.0	29.5	39.1
N	81.9	56.1	46.7	28.3	40.4
No translation features					
none	80.5	53.5	42.1	27.8	34.0
Only translation features					
only	73.1	51.1	50.4	32.5	43.8

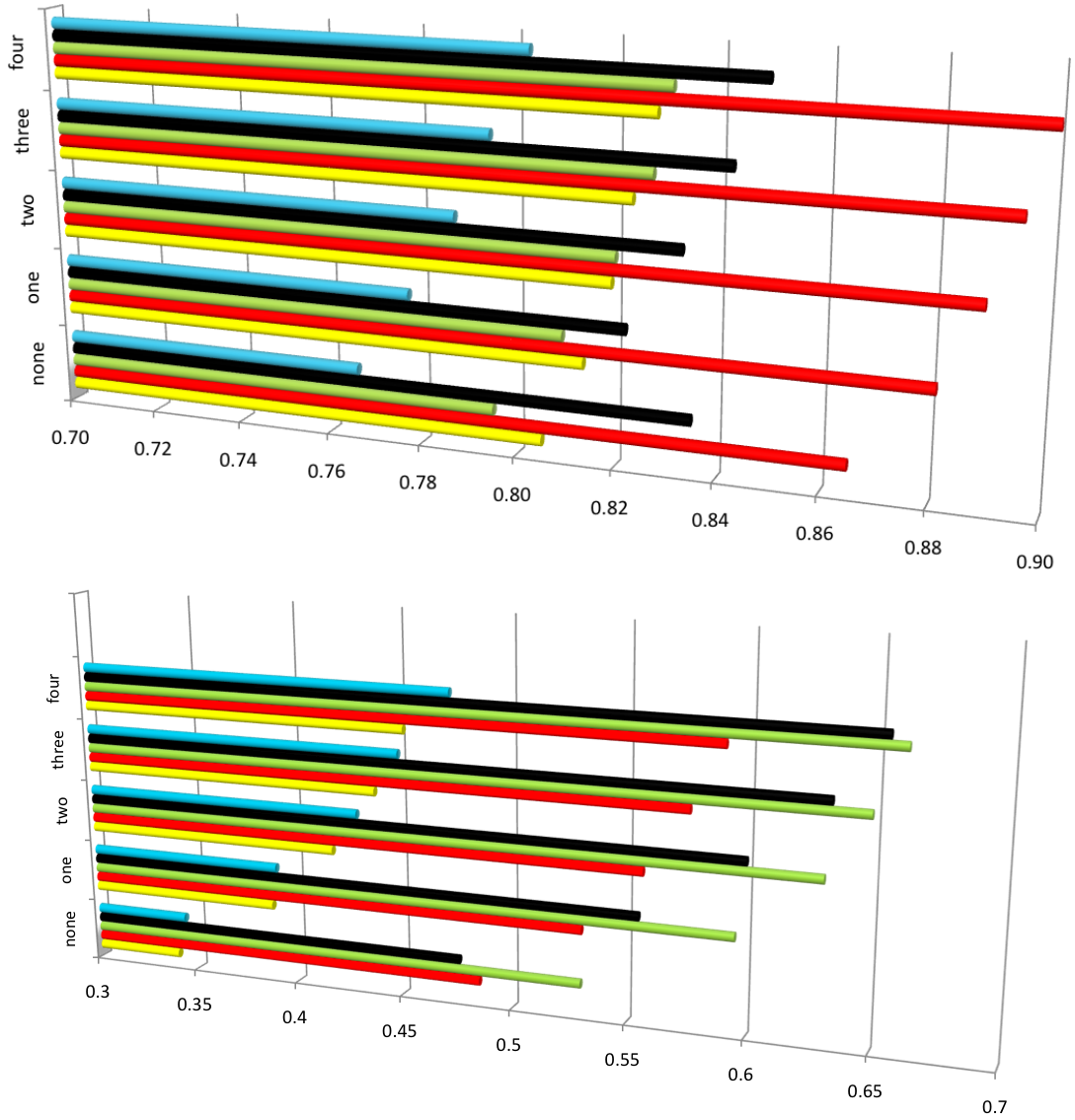


Fig. 2. Classification results for "bank" and "plant" for each of the target languages. The languages are resp. from top to bottom: Dutch, French, Italian, Spanish and German.

feature base level (e.g. by adding bag of word features for a broader context, etc.).

The scores clearly confirm the validity of our hypothesis: the experiments using all different translations as features are constantly better than the ones using less or no multilingual evidence. This conclusion holds for all five classification results. In addition, the scores also degrade relatively to the number of translation features that is used. This allows us to conclude that incorporating multilingual information in the feature vectors helps the classifier to choose more reliable and finer sense distinctions, which results in better translations in our case. Moreover, the more translations (in different languages) are incorporated in the feature vector, the better the classification results get. Another striking observation is that the classifier that solely relies on translation features (*Only translation features*) often beats the classifier that incorporates all context and translation features. There are, however, two limitations to our experimental framework. We have not experimented with a higher number of languages, and as a consequence we can not estimate from which number of languages the performance would start to degrade. In addition, another interesting line of research would be to include languages belonging to more distant language families.

The experimental results also reveal remarkable differences between the different languages. This can probably be explained by the difference in morphological structure between the two language families. As Dutch and German tend to concatenate the parts of compounds in one orthographic unit, whereas the romance languages (French, Italian, Spanish) keep these parts separated by spaces, this often results in compound

translations in German and Dutch. As a result, the number of different classes this classifier has to choose from, is much larger (as already shown in Figure 1). This difference is also reflected in the baselines, where the French, Italian and Spanish baseline is clearly higher than the Dutch or German one for most words.

Another interesting observation to make is that languages from the same language branch seem to contribute more to a correct classification result. The results show for instance that for the Spanish classifier, the use of Italian and French translations in the feature vector results in better classification scores, whereas for German, the incorporation of the Dutch translations in the feature vector seems to contribute most for choosing a correct translation. More experiments with other words and languages will allow us to examine whether this trend can be confirmed. Previous research on this topic has ended in contradictory results: Ide [18] showed that there was no relationship between sense discrimination and language distance, whereas Resnik and Yarowsky [6] found that languages from other language families tend to lexicalize more sense distinctions.

Our results clearly show that adding more multilingual evidence to the feature vector helps the WSD classifier to predict more accurate translations. The logical next step is to integrate this multilingual information into a real WSD application. In order to do so we will use the multilingual evidence from the parallel corpus to enrich our training vectors. Instead of only incorporating the aligned translations from the other languages, we will add all content words from the aligned translations as bag-of-word features to the feature vector. We will also develop a strategy to generate the corresponding translation features for the test instances. Both the local context features of the English target word and the cross-lingual evidence will be taken into account for computing the similarity scores between the test input and the training instance base. The expected outcome, based on the results we showed in this paper, is that each language can contribute to make finer sense distinctions and thus to provide more contextually accurate translations for the ambiguous target words.

V. Conclusion and Future Work

We presented preliminary results for a multilingual Word Sense Disambiguation system, which does not use labels from a predefined sense inventory, but translations that are retrieved by running word alignment on a parallel corpus. Although there is still a lot of room for improvement on the feature base, the scores of all five WSD systems constantly beat the most frequent translation baseline. The results allow us to develop a proof of concept that multilingual evidence in the feature vector, helps the classifier to make more reliable and finer sense distinctions, which result in better translations. We also observed that adding translations from the same language branch seems to help the classifier best to predict a correct translation in the focus language.

In future work, we want to run additional experiments with different classifiers on a larger sample of ambiguous words. We also wish to improve the classification results by performing joint feature selection and parameter optimization per ambiguous target word (E.g. by using a genetic algorithm approach). In addition, we also plan to include more multi-lingual evidence in a real WSD set-up. Therefore we will store the bag-of-words translation features resulting from the aligned translations in the training feature vectors, and add the automatically generated corresponding translation features for the test sentences to the test feature vectors.

References

[1] E. Agirre and P. Edmonds, Eds., *Word Sense Disambiguation*, ser. Text, Speech and Language Technology. Dordrecht: Springer, 2006.

[2] R. Navigli, "Word sense disambiguation: a survey," in *ACM Computing Surveys*, 2009, vol. 41, no. 2, pp. 1–69.

[3] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[4] A. Otegi, E. Agirre, and G. Rigau, "Ixa at clef 2008 robust-wsd task: Using word sense disambiguation for (cross lingual) information retrieval," in *Evaluating Systems for Multilingual and Multimodal Information Access 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17-19, 2008.*, 2009.

[5] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the MT Summit*, 2005.

[6] P. Resnik and D. Yarowsky, "Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation," *Natural Language Engineering*, vol. 5, no. 3, pp. 113–133, 2000.

[7] N. Ide, T. Erjavec, and D. Tufis, "Sense discrimination with parallel corpora," in *Proceedings of ACL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, PA, 2002, pp. 54–60.

[8] W. Gale, K. Church, and D. Yarowsky, "A method for disambiguating word senses in a large corpus," in *Computers and the Humanities*, 1993, vol. 26, pp. 415–439.

[9] H. Ng, B. Wang, and Y. Chan, "Exploiting parallel texts for word sense disambiguation: An empirical study," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Santa Cruz, 2003, pp. 455–462.

[10] M. Diab and P. Resnik, "An unsupervised method for word sense tagging using parallel corpora," in *Proceedings of ACL*, 2002, pp. 255–262.

[11] D. Tufiş, R. Ion, and N. Ide, "Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*. Geneva, Switzerland: Association for Computational Linguistics, Aug. 2004, pp. 1312–1318.

[12] Y. Chan and H. Ng, "Scaling up word sense disambiguation via parallel texts," in *AAAI'05: Proceedings of the 20th national conference on Artificial intelligence*. AAAI Press, 2005, pp. 1037–1042.

[13] W. Gale and K. Church, "A program for aligning sentences in bilingual corpora," in *Computational Linguistics*, 1991, pp. 177–184.

[14] F. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[15] W. Daelemans and A. van den Bosch, *Memory-Based Language Processing*. Cambridge University Press, 2005.

[16] H. Schütze, "Automatic word sense discrimination," *Computational Linguistics*, vol. 24, no. 1, pp. 97–123, 1998.

[17] A. Purandare and T. Pedersen, "Word sense discrimination by clustering contexts in vector and similarity spaces," in *Proceedings of the Conference on Computational Natural Language Learning*, 2004, pp. 41–48.

[18] N. Ide, "Parallel translations as sense discriminators," in *SIGLEX Workshop On Standardizing Lexical Resources*, 1999.

[19] W. Daelemans, J. Zavrel, and K. v. d. B. van der Sloot, "Timbl: Tilburg memory-based learner, version 4.3, reference guide," Tilburg University, Tech. Rep. ILK Technical Report - ILK 02-10, 2002.

[20] V. Hoste, I. Hendrickx, W. Daelemans, and A. van den Bosch, "Parameter optimization for machine-learning of word sense disambiguation," *Natural Language Engineering, Special Issue on Word Sense Disambiguation Systems*, vol. 8, pp. 311–325, 2002.
[21] J. Quinlan, *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.
[22] W. Daelemans, V. Hoste, F. De Meulder, and B. Naudts, "Combined optimization of feature selection and algorithm parameter interaction in machine learning of language," in *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, 2003, pp. 84–95.

Knowledge Expansion of a Statistical Machine Translation System using Morphological Resources

Marco Turchi and Maud Ehrmann

***Abstract*—Translation capability of a Phrase-Based Statistical Machine Translation (PBSMT) system mostly depends on parallel data and phrases that are not present in the training data are not correctly translated. This paper describes a method that efficiently expands the existing knowledge of a PBSMT system without adding more parallel data but using external morphological resources. A set of new phrase associations is added to translation and reordering models; each of them corresponds to a morphological variation of the source/target/both phrases of an existing association. New associations are generated using a string similarity score based on morphosyntactic information. We tested our approach on En-Fr and Fr-En translations and results showed improvements of the performance in terms of automatic scores (BLEU and Meteor) and reduction of out-of-vocabulary (OOV) words. We believe that our knowledge expansion framework is generic and could be used to add different types of information to the model.**

***Index Terms*—Machine translation, knowledge, morphological resources.**

I. Introduction

THE translation capability of a Statistical Machine Translation (SMT) system is driven by the training data and process. Big amounts of parallel data are used to allow the system to cover the source language as much as possible, but this effort collides with the vocabulary dimension of a language and the fact that the probability of finding unseen words in a language never vanishes. The inner knowledge of a system is the output of the training process that transforms the parallel data into tables: translation, language and reordering. Each item in translation and reordering tables associates textual (links phrase/s in different languages) and probability information (measures how reliable the information in the textual part is).

In real world translation systems, where source sentences may come from different domains, lack of knowledge is often responsible for translation quality: large number of OOV words or incorrect translations in target sentences are the main problems. In particular, when the source language is morphologically richer than the target language, translations are highly affected by the presence of OOV words. The other way around, the number of source phrases covered during the translation is higher, but target sentences contain more incorrect translated words.

Adding more data is the most obvious solution, but this has well-known drawbacks: it heavily increases the dimension of the tables, which reduces the translation speed, and parallel data are not always available for all the language pairs. In case of low quality parallel data, it can be even harmful because more data imply a bigger number of unreliable or incorrect associations built during the training phase.

In this paper, we address the problem of expanding the knowledge of an SMT system without adding parallel data, but extending the knowledge produced during the training phase. The main idea consists of inserting artificial entries in the phrase and reordering models using external morphological resources; the goal is to provide more translation options to the system during the construction of the target sentence.

Given an association of the phrase table, we first expand the source and target phrases, generating all their possible morphological variations. Then, given two sets of filtered new phrases in different languages, new associations are built computing the similarity between each element of the sets. Our similarity does not take into account the word forms but the morphosyntactic information of each token of the phrase. New associations are added to the phrase and reordering models multiplying the probabilities of the original association by the similarity score: most reliable associations get the highest scores. We test the expanded models on En-Fr and Fr-En translations using two different test sets and results show improvements of the performance in terms of Bleu [18], Meteor [15] and OOV word reduction and better translation of known phrases.

This paper is structured as follows: section II reports previous work, section III describes our expansion method, section IV sets the experimental framework, section V presents the results and, finally, section VI concludes and discusses future work.

II. Related Work

A large number of work has recently been proposed to increase the knowledge of an SMT system using external resources.

Manuscript received November 2, 2010. Manuscript accepted for publication January 14, 2011.

The authors are with the Joint Research Centre (JRC), IPSC - GlobSec, European Commission, Via Fermi 2749, 21027, Ispra (VA), Italy (e-mail: name.surname@jrc.ec.europa.eu.)

A classical approach consists of adding parallel data. In [20], the authors study the translation capability of a PBSMT system under different conditions, showing that the performance does not necessary improve when adding independent and identically distributed parallel data. They also suggest the generation of artificial training data based on existing training data, or *a posteriori* expansion of the tables. We follow these suggestions in our work. Other kind of parallel data can be used: in [19], parallel treebank data are added to a PBSMT system trained with Europarl data. Different approaches to incorporate such new data are proposed. They show that it is possible to raise the translation performance but, increasing the Europarl seed, the contribution of the treebank data decreases.

The knowledge of a PBSMT system can also be increased extracting different types of information from the training data and using all of them together. Koehn and Hoang [13] integrate additional annotations at the word level such as lemma, part-of-speech and morphological features. The proposed method outperforms the baseline in terms of automatic score and grammatical coherence.

Another approach consists in using some external data (monolingual or multilingual) to increase the existing knowledge; several methods have been proposed. Our selection may be representative but not exhaustive. Marton *et al.* [16] investigate how to augment training data by deriving monolingual paraphrases that are similar (in terms of distributional profiles) to OOV words and phrases, using distributional semantic similarity measures. Mirkin *et al.* [17] also propose an entailment-based approach to handle unknown words, using a source-language monolingual resource (WordNet) and a set of textual entailment rules. Both approaches show better results compared to the baseline. Haffari *et al.* [9] propose an active learning framework and try several sentence selection strategies, showing results accordingly. In [6], Garcia *et al.* propose to use a multilingual lexical database to compute more informed translation probabilities, showing good results when applying the MT system to a new domain.

Regarding the use of morphology in the SMT, a lot of work has been done (see Yang and Kirchhoff [21]), but few of it has analysed directly the phrase table content. When encountering unseen verbal forms, De Gispert *et al.* [3] look for similar known forms and generate new phrases on the source and target sides, using morphological and shallow syntax information. With this method, they show improvements in terms of Bleu score. Yang and Kirchhoff [21] propose a hierarchical backoff model based on morphological information: for an unseen word, the model relies on translation probabilities derived from stemmed or split versions of the word. Habash [8] uses morphological inflection rules to match OOV words with INV (in vocabulary) words and to generate new phrases in which INV words are replaced by OOV words. In his experiments, this approach allows the system to handle 60% of the OOV.

In this paper, we propose a morphologically-based method to expand the existing knowledge of an SMT system. This new knowledge is then used by the PBSMT system to handle unseen words and to produce more reliable translations for seen words. As far as we know, this is the first attempt to generate new high quality associations using morphological resources and considering *all* original associations in the phrase table, whatever their part of speech is.

III. Knowledge Expansion

In this work, we focus our attention on the fact that, in an SMT system, each word form is treated as a token: two words, one morphological variation of the other, are different and independent tokens. Therefore, if one of the morphologically-related word forms is not in the training data, the word will become an OOV word or will be wrongly translated. Let's consider an example, from French to English:
Source: . . . *les élections parlementaires anticipées en autriche ont apporté un affaiblissement sensible de la principale coalition* . . .
Target: . . . *the early parliamentary elections in austria have apporté*|||UNK *a weakening sensitive of the principal coalition* . . .
In the translated sentence, the word *apporté* is not translated (marked as unknown) and the word *principale* is translated as *principal* instead of *leading* (as it is in the reference sentence), even if in the translation phrase table learned during the training phase we have the following associations[1]:

```
apporte ||| brings ### apporte ||| provides ### nous apportons
||| we provide ### principale ||| principal ### principales
||| leading
```

Our approach proposes to use morphological resources to expand the knowledge of the system: new associations are generated and added to the phrase and reordering models; these new associations contain morphological variations of source and target phrases created during the training process. Regarding the previous example, the phrase table (PT) will be expanded with the associations `apporté ||| brought` and `principale||| leading`, enabling the SMT system to correctly translate the sentence.

The process of generation of new associations takes as input the phrase and reordering tables on one side, and morphological resources on the other. In our experiments we used the English and French Multext morphological resources [4]. These morphosyntactic lexicons provide, for each lexical entry, three types of information: the word form (*brought*), its lemma (*bring*), and finally its MorphoSyntactic Description (MSD, *Vviq3s*). The MSD is a condensed tag that encodes the morphosyntactic features of the word, in the form of attribute-value pairs specified via letters (part of speech, gender, number, tense, mood, etc.). One significant advantage of Multext resources is that they provide harmonized morphosyntactic description for more than 15

[1]Only the textual part is presented here.

languages. The whole chain of knowledge expansion is made up of five steps, described in the next sections.

Monolingual Expansion of a Phrase. Given an association from the PT, the first objective is to generate all the possible morphological variations for each of its monolingual parts. For each token of a monolingual phrase, we first generate a vector that contains all its morphological variations. To do so, we look for its associated lemma(s) in the morphological resources and return all the words that share this lemma. We then apply a recursive algorithm that takes, for each phrase, the morphological variation vector and produces new phrases, in which each token is associated with its MSD. This monolingual expansion phase is done for all the tokens, whatever their part of speech (POS) is. In our example, if we take the phrase `nous apportons`, we first expand "nous" then "apportons" and finally we build new phrases, as illustrated in Figure 1. Due to the absence of any constraints in this phrase expansion step, wrong phrases are generated (marked with stars in Figure 1). A filtering step is therefore needed.

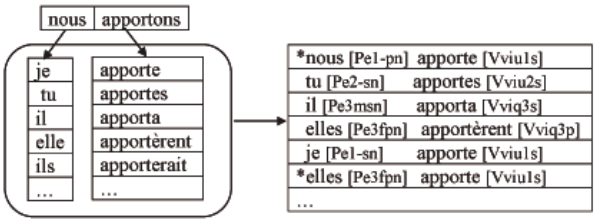


Fig. 1. Example of a monolingual phrase expansion.

Phrase Filtering. We defined two kinds of filtering. The first one, based on probability checking, is designed to carry out a coarse selection. The second one, based on grammatical rules, performs a fine-grained selection.

Probability checking filtering takes advantage of the language models created using more than 3 million sentence pairs. For each language, the language model is queried with the generated phrases and then probabilities of correctness are computed for each phrase. The list of phrases is sorted according to the probabilities and only phrases above a defined threshold are kept. This threshold was computed using human-annotated data: given a randomly selected set of phrases for each phrase length (from 1 to 7 tokens) and for each language, phrases were expanded and manually annotated according to their grammatical and semantic correctness. Thousands of new phrases were annotated for English and French. These phrases were then sorted according to their probabilities (computed against the LM) and, for each possible threshold value, the $F_{0.5}$ score was calculated in reference to annotated data. We used the $F_{0.5}$ score because it weights precision twice as much as recall, and we prefer to generate good quality data, even if there are less new associations. For each language and phrase length, we computed the maximum $F_{0.5}$ score values and took their relative threshold values. Figure 2 illustrates the threshold computation for English phrase lengths.

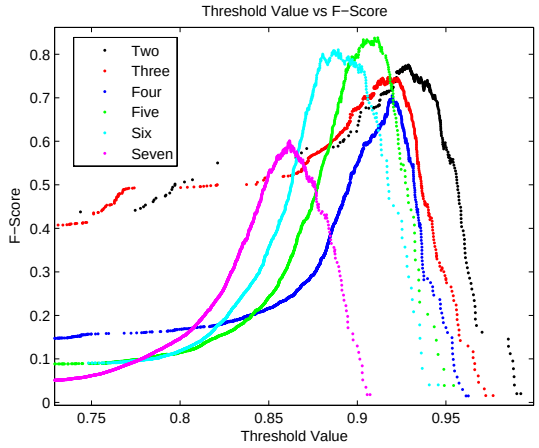


Fig. 2. Computation of the English thresholds by phrase length.

This language model filtering approach is able to remove an important number of wrong phrases; however, it is data-dependent and does not filter out all the unwanted phrases. To supplement this first coarse filtering, we use a set of hand-written grammatical rules based on morphosyntactic tags. The rules allow to verify gender and number agreement between tokens, to check tense agreements within verbal phrases, and/or to remove obvious wrong sequences of tags (like three or four times the same POS tag in a row). This second filtering gives, again, the opportunity to remove wrong phrases from the list. At this stage we have, for each language, a list of correct new phrases (in Figure 1, only phrases without stars remain in the list). These new phrase lists are then used to produce new associations.

Generation of New Associations. The objective of this step is, given two sets of new phrases, to create new associations. To match phrases, we use a string matching similarity score based on morphosyntactic information. If we consider the PT association `nous apportons|||we bring|||(0)(1)|||(0)(1)` (numbers state the word alignment), first steps should have produced two lists of correct new phrases, where each token is associated with its MSD, (`il[Pe3msn] apporta[Vviq3s]` and `we[Pe1-pn] brought[Vviq2p]`).

Given two phrases (p_1 and p_2) in different languages, the morphosyntactic descriptions of their tokens (t_1^{msd} and t_2^{msd}) and the number of elements in the word alignment (a) of the original association, we compute the similarity as:

$$s(p_1, p_2) = \frac{\sum_{i,j \in a} st(t_i^{msd}, t_j^{msd})}{|a|} \quad (1)$$

TABLE I
EXAMPLES OF MSD SIMILARITY SCORE COMPUTATION

	Vviq3s and **Vviq2p**						**Pe3msn** and **Pe1-pn**					
MSD1	v	i	q	3	s		e	3	m	s	n	
MSD2	v	i	q	2	p		e	1	-	p	n	
Score	1	1	1	0	0	3/5	1	0	0,5	0	1	2.5/5

where:

$$st(t_i^{msd}, t_j^{msd}) = \frac{\sum_{i \in len(t_i^{msd})} m(t_i^{msd}(i), t_j^{msd}(j))}{len(t_i^{msd})} \quad (2)$$

The similarity between two phrases corresponds to the sum of the similarities between two tokens, normalized by the numbers of aligned tokens in the original associations (1); then, the similarity between two tokens corresponds to the similarity between two morphosyntactic descriptions given a matrix *m*, normalized by the length of the MSD (2). Considering the two new phrases generated from the PT association, `il[Pe3msn] apporta[Vviq3s]` and `we[Pe1-pn] brought[Vviq2p]`, the similarity between these phrases is equal to the similarity between the MSD "Pe3msn" (from *il*) and the MSD "Pe1-pn" (from *we*) plus the similarity between the MSD "Vviq3s" (from *apporta*) and the MSD "Vviq2p" (from *brought*), all divided by 2, which corresponds to the number of elements in the original association alignment (`(0)(1)|||(0)(1)`). In case of multi-alignment, the similarity of the single token is computed against all its aligned tokens.

The similarity between MSDs corresponds to a positional score based on a substitution matrix: each entry in the matrix describes the rate at which one character (in our case a letter encoding morphosyntactic information) in a MSD can be changed to another. Matrices were manually built by a linguist for the following parts of speech: Noun, Verb, Adjective, Pronoun, Determiner, Adverb, Preposition, Conjunction and Numeral. Within matrices, we decided to use the following values: 0 for morphological information that should not be matched (singular with plural for example), 0.5 for information that can be matched but not necessarily (feminine with neutral) and 1 when information should be matched (present tense with present tense). Regarding our example, the similarities between `apporta[Vviq3s] brought[Vviq2p]` and `il[Pe3msn] we[Pe1-pn]` are illustrated in Table I. Single character scores are obtained querying the Verb and Pronoun matrices (V and P).

For all potential phrase associations from the filtered lists of expanded phrases, we computed the similarity as described above. We then ranked the associations by similarity and computed a threshold corresponding to: $max - (max * 10\%)$, *max* being the maximum similarity value of the new association set. We finally keep the associations which have similarity values bigger than this threshold. In our example, the similarity between the two phrases is $\frac{\frac{2,5}{5}+\frac{3}{5}}{2} = 0.55$. If we have the same MSDs in both phrases, the maximum reachable would be 1 and the relative threshold is 0.9. In this case, the new association would be discarded. At the end, we have at our disposal "artificial" new associations that can be added to the phrase and reordering tables. Before doing so, we completed an evaluation of the new associations.

TABLE II
MANUAL EVALUATION OF NEW ASSOCIATIONS GENERATED EXPANDING 1,000 RANDOM PT ENTRIES

Alignment Type	**Precision**	**Number of New Associations**
A	0.6725	1933
no M	0.7544	1820
no M + E	0.8261	1530
no M + E + OE	**0.8861**	1115

TABLE III
NUMBER OF ENTRIES IN THE PHRASE TABLE

	Fr-En (News)	**En-Fr (News)**	**En-Fr (Europ.)**
Original	3,946,143	3,924,804	60,873,395
Reduced	229,390	217,685	4,480,135
Expanded	345,896	334,188	5,671,418

New Association evaluation. To evaluate the new associations, we randomly selected 1,000 associations from the Fr-En phrase table, expanded them using our algorithm and manually annotated.[2] The manual annotation was done in rather a strict way: an association was considered as correct if there was no mistake, neither in the phrases, nor in the association. Regarding the original association we did not judge its quality but we took into account different types of alignment. We distinguish between the following cases: *multi-alignment* (M), when a token on one side is aligned with several on the other ((0)(0)(0)…|(0,1,2)…), *one empty alignment* (OE), when one token on one side does not have a correspondence on the other ((0)() |(0)), and *several empty alignments* (E), when more than one token on one side does not have corresponding tokens on the other (()(1)()(0)()()() | (3)(1)). We computed the Precision according to these different types.

Results are presented in Table II. Precision is affected by two phenomena: the type of alignment taken into account and the phrase length (results by phrase length are omitted due to lack of space). Essentially, the measure increases removing multi and empty alignments (we add less new associations but of better quality), and considering shorter phrases. Showing up cases where new associations are of better or lower quality, this evaluation helped us to decide which type of original association to expand. The next section considers how to add new associations to the model.

Integration of New Associations. Starting from the PT, we artificially generate new associations that are finally added to the phrase and reordering models which constitute, at the end, an extended model. While adding new data to the original tables, we pay attention to do so respecting the way the data

[2]As the expansion process is symmetric, the evaluation from the fr-en phrase table is also valid for the en-fr one.

were produced. New associations are made of three parts: a textual part (the newly generated phrases), a "word order" part (we keep the same as the original association), and a probability part, that indicates how reliable an association is. Probabilities taken in account are bidirectional translation probabilities and lexical weighing for the phrase table and bidirectional monotone, swapped and discontinuous reordering probabilities in the reordering model. In our extended model, the probability of a new association is computed multiplying the probabilities of the original association by the similarity score of the new association. This allows two things: at the phrase table level, original associations get the highest probabilities; then, within a set of new associations generated from a particular PT association, each new association has its own probability, reflecting how reliable its generation process was. In this way, phrase and reordering tables can be extended without perturbing the original knowledge of the system. Finally, if a new association is a duplicate of an original one, it is not added to the new model.

IV. Experimental Setting

To assess our approach, we conducted a series of experiments on French-to-English and English-to-French translations. They were run using Moses [14], a complete phrase-based machine translation toolkit for academic purposes, and IRSTLM [5] for language modelling during the phrase filtering and the pure translation. Results have been evaluated in terms of Bleu and Meteor scores over lowercased output and number of OOV words. Meteor considers the surface form of each word and does not make use of WordNet synonyms.

SMT data. We trained the PBSMT models using two different training corpora: Commentary English-French and French-English News corpus [1] containing 64,233 sentence pairs in both directions and Europarl Release v3 English-French [12] containing 1,428,799 sentence pairs. We used two test sets in both language pair directions coming from different domains: 3,000 sentences from Commentary News and 2,000 from the proceedings of Europarl, both selected by the organizers of the Statistical Machine Translation Workshop [1].

Pre-Processing of the PT. The translation table contains all phrase pairs found in the training corpus, which includes a lot of noise. Our approach expands all the associations found in the translation model without taking into account their correctness. To avoid the expansion of unreliable associations, we pre-processed the phrase table using the method proposed by Johnson et al. [10]. This approach prunes the PT using a technique based on the significance testing of phrase pair co-occurrence in the parallel corpus. In our experiments we used a threshold equal to $\alpha + \epsilon$ and only the top 30 phrase translations for each source phrase based on $p(e|f)$ were kept. If a phrase pair appears exactly once in the corpus and each of the component phrases occurs exactly once on its side of the parallel corpus, this special case is called 1-1-1 association. Our parameter choices removed all the 1-1-1 associations.

New PT dimensions resulting from this pruning process are shown in Table III. Testing the original and reduced models on the test data confirmed the results found in [10]: substantial reduction of the PT dimension does not alter the translation performance. In the rest of the paper, baseline results refer to the performance of the reduced model.

V. Results

Three PBSMT systems were built using the training data presented above: French-English translation trained on the Commentary News data ($F2E_N$), English-French translation trained on the Commentary News data ($E2F_N$) and English-French translation trained on the Europarl data ($E2F_E$). For each of them, phrase tables were pre-processed and then expanded using our algorithm. According to what we learnt from the evaluation of the new associations (section III), we expanded original associations that do not contain multi or empty (one or more than one) alignments. Even if this choice reduced the number of new associations, it guarantees high precision in what is added (Table II).

The expansion of phrase and reordering models requires a counterpart information in the language model. Thus, a language model was created using the target side of the training data plus an external corpus crawled on the Web containing 3,463,954 French and 3,183,871 English sentences. Performance of the baseline and extended models is shown on the left side of Table IV. In all experiments, the number of OOV words decreases; this is more evident in Fr-En translation, as the source language is more morphologically inflected. In terms of automatic scores, the $F2E_N$ and $E2F_N$ expanded models resulted in improvements with respect to the baseline.

Knowledge expansion should allow the model not only to translate unknown words (in our initial example, *apporté* is translated into *brought*) but also to better translate already known ones (*principal* is replaced by *leading*). In order to evaluate this phenomenon, we conducted a manual evaluation on a set of 110 randomly selected target sentences ($F2E_N$) where there is a difference (increase or decrease) in Meteor score between the baseline and expanded system translations. Comparing them, we distinguished several causes of score variation: unknown word covered (Unknown), known word substituted (Known), unknown word covered and known word substituted (Both) and other reasons like word reordering (Other). The results of this manual evaluation (Table V) confirm that the expanded model performs better than the baseline and show that improvements not only come from unknown word coverage but also from better translations of known words.

From a manual analysis of the $F2E_N$ translated sentences, we additionally noticed that in several cases the automatic scores are not able to capture improvements given by the expanded models, see [2] for more details on this problem.

TABLE IV
OBTAINED RESULTS

	3-gram Language Model				2-gram Language Model (test set)			
	Commentary News		Europarl		Commentary News		Europarl	
Fr-En Commentary News ($F2E_N$)								
	Baseline	Expanded	Baseline	Expanded	Baseline	Expanded	Baseline	Expanded
Bleu %	21.68	21.89	21.99	**22.37** *	26.41	27.01 *	26.46	**27.17** *
Meteor	0.4698	**0.4733** *	0.4706	0.4720	0.4975	0.5035 *	0.4972	**0.5042** *
OOV	7,763	**7,004**	3,107	2,741	7,763	**7,004**	3,107	2,741
En-Fr Commentary News ($E2F_N$)								
Bleu %	21.35	21.61 *	23.62	**23.79** *	24.66	**25.22** *	25.89	26.36 *
Meteor	0.1524	0.1542 *	0.1630	**0.1650** *	0.1739	0.1780 *	0.1805	0.1842 *
OOV	6,447	**5,977**	2,400	2,153	6,447	**5,977**	2,400	2,153
En-Fr Europarl ($E2F_E$)								
Bleu %	22.62	**22.63**	27.43	27.38	28.51	**28.73** *	34.75	34.77
Meteor	0.1608	0.1607	0.1927	0.1923	0.2025	0.2040 *	0.2465	0.2467
OOV	3,357	**3,186**	260	253	3,357	**3,186**	260	253

TABLE V
HUMAN EVALUATION OF A SAMPLE OF 110 RANDOM SELECTED SENTENCES FROM $E2F_N$

	Total	Unknown	Known	Both	Other
Increment in Meteor	84	18	45	2	19
		21.4%	53.5%	2.3%	22.6%
Decrement in Meteor	26	0	16	0	10
		0	61.5%	0	38.5%

BASELINE: *we have settled our divergentes|||UNK views ...*
EXPANDED: *we have settled our divergent views ...*
REFERENCE: *we've resolved our differing opinions ...*

In this example, the expanded sentence has no OOV words and is more comprehensible for a non-French speaker, but there is not improvement regarding the automatic scores. This kind of example, combined with the need of a counterpart in the language model, raised the following question: Was the correct translation of the word *divergentes* – according to the reference sentence – present in the model?

Controlled environment experiments. To answer these questions, we ran a set of controlled environment experiments. Our idea was to evaluate only the knowledge of the phrase and reordering models cutting out the language model contribution. Instead of using the big language model, which obviously was not exhaustive and could negatively influence the performance, we used a 2-gram language model built on the target side of the test set. Regardless of the small number of sentences used and of the fact that probabilities may not be accurately estimated, it drove the decoder to select those phrases that were present in the reference sentences. Differences in performance between the baseline and expanded models reflect only the difference in terms of knowledge in the phrase and reordering tables. Results are shown on the right side of Table IV.

Results in the Table IV are obtained using a 3-gram language model trained on the target side of the training data plus 3,463,954 French sentences or 3,183,871 English sentences. * = significance test over baseline with $p < 0.0001$, using pair-wise bootstrap test with 95% confidence interval [11]

In these controlled environment experiments, the gap between the baseline and the expanded models increased with a maximum 0.73 Blue score points. The augmented system has a significant gain over its baseline also in the $E2F_E$ translations using the out-of-domain test set. These results show how the new model took advantage of the information added by the new associations, increasing the quality of the output translations. This means that the new model has the correct information to produce a target sentence similar to the reference sentence, but the selection of the correct translation option is strictly related to the language model information. Target sentences that are not similar to the reference sentences are not necessarily wrong.

VI. DISCUSSION AND FUTURE WORK

This work shows that the knowledge of a Statistical Machine Translation system can be artificially expanded without relying on parallel data. Morphological resources are used to generate new high quality associations that are added to phrase and reordering models. Each new association contains source/target/both phrases that are morphological variations of the original ones. Although this may be considered a limitation, because "never seen" associations cannot be added, results confirm the benefits in terms of translation quality.

Our algorithm increases the dimension of the PTs (see Table III): for models trained with Commentary News roughly about 50%, while for the Europarl model about 25%. This assumes particular relevance if we thought that in the reduced tables 1-1-1 associations are pruned, see Section IV. It means that each new association that the proposed method adds would require at least more than one parallel sentence pairs to be added during the training phase using parallel data.

Empirical results support the assumption that the new associations help the SMT system to better translate sentences coming from different domains. Our expanded models performed better than the baseline in particular when the original model is trained on a small training set. It reduces the impact of the OOV words in the translation, but not only:

manual evaluation shows that also known words are replaced by better translations in the target sentences.

Manual analysis of the results highlighted the weakness of the automatic score to catch improvements in translation. This suggested to us a series of experiments with a small but optimal language model. In this controlled environment, results show even more benefits of our approach with any different training set sizes and test data. This confirms that extra knowledge can be used by the decoder only with a language model that contains suitable information.

Our intention is to make our technique more portable to other language pairs replacing the grammatical rules with a language model built on part of speech information. The idea of expanding the knowledge of an SMT system is generic and different types of information can be passed artificially to it. In this paper we investigated how to add morphologically related new associations; in a next step, we will consider how to add new semantically related associations, e.g. semantic knowledge. We believe that the benefits of our approach will be more evident using more inflected languages like Czech. Experiments are planed in this direction.

ACKNOWLEDGEMENT

The authors would like to thank Nello Cristianini for preliminary discussion about this work, Ralf Steinberger and Jenya Belyaeva for comments and suggestions that helped to improve the present paper.

REFERENCES

[1] C. Callison-Burch, P. Koehn, C. Monz, and J. Schroeder, "Findings of the 2009 Workshop on Statistical Machine Translation," in *Proceedings of WSMT*, 2009, pp. 1-28.

[2] C. Callison-Burch and M. Osborne, "Re-evaluating the role of BLEU in machine translation research," in *Proceedings of EACL*, 2006, pp. 249-256.

[3] A. De Gispert, J.B. Mariño, and J.M. Crego, "Improving statistical machine translation by classifying and generalizing inflected verb forms," in *Proceedings of 9th European Conference on Speech Communication and Technology*, 2005, pp. 3193-3196.

[4] T. Erjavec, "MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora," in *Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation*, 2004.

[5] M. Federico, N. Bertoldi, and M. Cettolo, "Irstlm: an open source toolkit for handling large scale language models," in *Proceedings of Interspeech*, 2008, pp. 1618-1621.

[6] M. Garcia, J. Giménez, and L. Màrquez, "Enriching Statistical Translation Models Using a Domain-Independent Multilingual Lexical Knowledge Base," *Lecture notes in computer science (Computational Linguistics and Intelligent Text Processing)*, vol. 5449, pp. 306-317, 2009.

[7] S. Goldwater and D. McClosky, "Improving statistical MT through morphological analysis," in *Proceedings of EMNLP*, 2006, pp. 676-683.

[8] N. Habash, "Four techniques for online handling of out-of-vocabulary words in Arabic-English statistical machine translation," in *Proceedings of ACL*, 2006, pp. 57-60.

[9] G. Haffari, M. Roy, and A. Sarkar, "Active learning for statistical phrase-based machine translation," in *Proceedings of NAACL*, 2009, pp. 415-423.

[10] H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," *Proceedings of EMNLP-CoNLL*, 2007, pp. 967-975.

[11] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of EMNLP*, 2005, pp. 388-395.

[12] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of MT summit*, 2005.

[13] P. Koehn and H. Hoang, "Factored translation models," in *Proceedings of EMNLP-CoNLL*, 2007, pp. 868-876.

[14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico and others, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL, demonstration session*, 2007, pp. 1618-1621.

[15] A. Lavie and A. Agarwal, "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, pp. 228-231.

[16] Y. Marton, C. Callison-Burch, and P. Resnik, "Improved statistical machine translation using monolingually-derived paraphrases," in *Proceedings of EMNLP*, 2009, pp. 381-390.

[17] S. Mirkin, L. Specia, N. Cancedda, I. Dagan, M. Dymetman, and I. Szpektor, "ource-language entailment modeling for translating unknown terms," in *Proceedings of ACL*, 2009, pp. 791-799.

[18] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL*, 2002, pp. 311-318.

[19] J. Tinsley, M. Hearne and A. Way, "Exploiting parallel treebanks to improve phrase-based statistical machine translation," in *Proceedings of CICLing*, 2009, pp. 318-331.

[20] M. Turchi, T. DeBie, and N. Cristianini, "Learning performance of a machine translation system: a statistical and computational analysis," *Proceedings of the Third Workshop on Statistical Machine Translation*, 2008, pp. 35-43.

[21] M. Yang and K. Kirchhoff, "Phrase-based backoff models for machine translation of highly inflected languages," in *Proceedings of EACL*, 2006, pp. 41-48.

Low Cost Construction of a Multilingual Lexicon from Bilingual Lists

Lian Tze Lim, Bali Ranaivo-Malançon, and Enya Kong Tang

***Abstract*—Manually constructing multilingual translation lexicons can be very costly, both in terms of time and human effort. Although there have been many efforts at (semi-)automatically merging bilingual machine readable dictionaries to produce a multilingual lexicon, most of these approaches place quite specific requirements on the input bilingual resources. Unfortunately, not all bilingual dictionaries fulfil these criteria, especially in the case of under-resourced language pairs. We describe a low cost method for constructing a multilingual lexicon using only simple lists of bilingual translation mappings. The method is especially suitable for under-resourced language pairs, as such bilingual resources are often freely available and easily obtainable from the Internet, or digitised from simple, conventional paper-based dictionaries. The precision of random samples of the resultant multilingual lexicon is around 0.70–0.82, while coverage for each language, precision and recall can be controlled by varying threshold values. Given the very simple input resources, our results are encouraging, especially in incorporating under-resourced languages into multilingual lexical resources.**

***Index Terms*—Lexical resources, multilingual lexicon, under-resourced languages.**

I. Introduction

MULTILINGUAL translation lexicons are very much desired in many natural language processing (NLP) applications, including multilingual machine translation and cross-lingual information retrieval, but are very costly to construct manually. On the other hand, given the abundance of bilingual machine readable dictionaries (MRD), there have been many efforts at (semi-)automatically merging these bilingual lexicons into a sense-distinguished multilingual lexicon [1]–[3].

Many of these approaches require the input bilingual lexicons to include certain types of information besides equivalents in the target language, such as gloss or definition text in the source language and domain field codes. Unfortunately, bilingual lexicons with such features are not always available, especially for under-resourced language pairs. Nor are the delineation or granularity of different sense entries indicated clearly or consistently. More often than not, the lowest common denominator across bilingual lexicons is just a simple list of mappings from a source language word to one or more target language equivalents.

Manuscript received November 2, 2010. Manuscript accepted for publication January 22, 2011.

The authors are with the Natural Language Processing Special Interest Group, Faculty of Information Technology, Multimedia University, Malaysia (e-mail: liantze@gmail.com, {ranaivo, enyakong}@mmu.edu.my).

English	Chinese	Malay	French
factory plant	工厂	loji kilang	fabrique manufacture usine

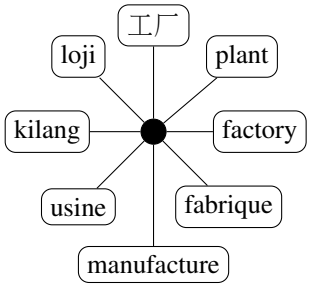


Fig. 1. Example multilingual lexicon entry for the concept *industrial plant* with lexical items from English, Chinese, Malay and French.

We aim to bootstrap a multilingual translation lexicon, given the simplest bilingual dictionaries taking the form of simple lists of bilingual translations. Such low resource requirements (as well as the low-cost method that will be described) is especially suitable for under-resourced language pairs. We first give a brief overview of the overall structure of the multilingual lexicon in Section II. Section III describes how an initial trilingual lexicon can be generated from bilingual ones, and how further languages can be added. Initial experimental results presented in Section IV show that our method is capable of generating a usable multilingual dictionary from simple bilingual resources without the need for rich information types, such as those mentioned in Section V.

II. Multilingual Lexicon Organisation

Each entry in our multilingual lexicon is similar to a *translation set* described by Sammer and Soderland [4] as 'a multilingual extension of a WordNet synset [5]' and contains 'one or more words in each k languages that all represent the same word sense'. Unlike Sammer and Soderland's translation sets, however, our lexicon entries currently do not include any gloss or contexts to indicate the intended word sense.

Figure 1 shows an example translation set entry representing the concept *industrial plant*, containing English '*factory*' and '*plant*'; Chinese '工厂' (*gōngchǎng*); Malay '*loji*' and '*kilang*'; French '*fabrique*', '*manufacture*' and '*usine*'.

Internally, each translation set is accessed by a language-independent axis node, with language-specific lexicalisations connected to it, similar to the structural scheme used in the multiligual extension of the Lexical Markup Framework [6] and the Papillon Multilingual Dictionary [7]. Our multilingual lexicon is thus capable of handling lexical gaps (when a concept is not lexicalised in a language) as well as diversification phenomena (when a word sense in a language is more specific than its translation in another language). Nevertheless, for our current experiment, we will allow diversified meanings to be connected directly to the same axis.

III. Building the Lexicon

Our bootstrapping algorithm first generates trilingual translation triples based on the one-time inverse consultation (OTIC) procedure [8], which was proposed to generate translation lexicons for new language pairs from existing bilingual lexicons. These triples are then merged to produce the translation sets in our multilingual lexicon. New languages are added by producing translation triples containing the new language and languages already present in our multilingual lexicon, then merging the new triples into the existing entries by detecting common translation pairs.

A. One-time Inverse Consultation

Tanaka, Umemura and Iwasaki [8] first proposed OTIC to generate a bilingual lexicon for a new language pair L_1–L_3 via an intermediate language L_2, given existing bilingual lexicons for language pairs L_1–L_2, L_2–L_3 and L_3–L_2. Following is an example of a OTIC procedure for linking Japanese words to their Malay translations via English:

- For every Japanese word, look up all English translations ($\mathbb{E}_1$).
- For every English translation, look up its Malay translations ($\mathbb{M}$).
- For every Malay translation, look up its English translations ($\mathbb{E}_2$), and see how many match those in $\mathbb{E}_1$.
- For each $m \in \mathbb{M}$, the more matches between $\mathbb{E}_1$ and $\mathbb{E}_2$, the better m is as a candidate translation of the original Japanese word.

$$\text{score}(m) = 2 \times \frac{|\mathbb{E}_1 \cap \mathbb{E}_2|}{|\mathbb{E}_1| + |\mathbb{E}_2|}$$

A worked example is shown in Figure 2. The Japanese word '印' (*shirushi*) has 3 English translations, which in turn yields another three Malay translations. Among them, '*tera*' has 4 English translation, 2 of which are also present in the earlier set of 3 English translations. The one-time inverse consultation score for '*tera*' is thus $2 \times \frac{2}{3+4} = 0.57$, and indicates '*tera*' is the most likely Malay translation for '印'.

Bond et. al. [10] extended OTIC by linking through two languages, as well as utilising semantic field code and classifier information to increase precision, but these extensions may not always be possible as not all lexical resources include these information (nor do all languages use classifiers).

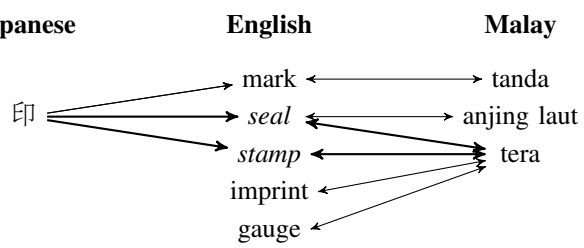


Fig. 2. Using OTIC, Malay '*tera*' is determined to be the most likely translation of Japanese '印' as they are linked by the most number of English words in both directions, with score('*tera*') $= 2 \times \frac{2}{3+4} = 0.57$. (Diagram taken from [9, Figure 1])

B. Extension to OTIC

OTIC was originally conceived to produce a list of bilingual translations for a new language pair. As our aim is a multilingual lexicon instead, we modified the OTIC procedure to produce trilingual translation triples and translation sets, as outlined in Algorithm 1.

Algorithm 1 allows partial word matches between the 'forward' and 'reverse' sets of intermediate language words. For example, if the 'forward' set contains '*coach*' and the reverse set contains '*sports coach*', the modified OTIC score is $\frac{1}{2} = 0.5$, instead of 0. This would also serve as a likelihood measure for detecting diversification in future improvements of the algorithm. The score computation for (w_h, w_t) is also adjusted accordingly to take into account this substring matching score (line 10), as opposed to the exact matching score in the original OTIC.

We retain the intermediate language words along with the 'head' and 'tail' languages, i.e. the OTIC procedure will output translation *triples* instead of pairs. α and β on line 14 are threshold weights to filter translation triples of sufficiently high scores. Bond et. al. [10] did not discard any translation pairs in their work; they left this task to the lexicographers who preferred to whittle down a large list rather than adding new translations. In our case, however, highly suspect translation triples must be discarded to ensure the merged multilingual entries are sufficiently accurate. Specifically, the problem is when an intermediate language word is polysemous. Erroneous translation triples (w_h, w_m, w_t) may then be generated (with lower scores), where the translation pair (w_h, w_m) does not reflect the same meaning as (w_m, w_t). If such triples are allowed to enter the merging phase, the generated multilingual entries would eventually contain words of different meanings from the various member languages: for example, English '*bold*', Chinese '黑体' (*hēitǐ* bold typeface) and Malay '*garang*' (fierce) might be placed in the same translation set by error.

As an example, consider the (w_h, w_m, w_t) translation triples with non-zero scores generated by OTIC where w_h = '*garang*',

Algorithm 1: Generating trilingual translation chains

for all lexical items $w_h \in L_1$ **do**
  $\mathbb{W}_m \leftarrow$ translations of w_h in L_2
  for all $w_m \in \mathbb{W}_m$ **do**
    $\mathbb{W}_t \leftarrow$ translations of w_m in L_3
    for all $w_t \in \mathbb{W}_t$ **do**
      Output a translation triple (w_h, w_m, w_t)
      $\mathbb{W}_{m_r} \leftarrow$ translations of w_t in L_2
      $$\text{score}(w_h, w_m, w_t) \leftarrow \sum_{w \in \mathbb{W}_m} \frac{\text{number of common words in } w_{m_r} \in \mathbb{W}_{m_r} \text{ and } w}{\text{number of words in } w_{m_r} \in \mathbb{W}_{m_r}}$$
    end for
    $$\text{score}(w_h, w_t) \leftarrow 2 \times \frac{\sum_{w \in \mathbb{W}_m} \text{score}(w_h, w, w_t)}{|\mathbb{W}_m| + |\mathbb{W}_{m_r}|}$$
  end for
  $X \leftarrow \max_{w_t \in \mathbb{W}_t} \text{score}(w_h, w_t)$
  for all distinct translation pairs (w_h, w_t) **do**
    if $\text{score}(w_h, w_t) \geq \alpha X$ or $(\text{score}(w_h, w_t))^2 \geq \beta X$ **then**
      Place $w_h \in L_1$, $w_m \in L_2$, $w_t \in L_3$ from all triples $(w_h, w_{...}, w_t)$ into same translation set
      Record $\text{score}(w_h, w_t)$ and $\text{score}(w_h, w_m, w_t)$
    else
      Discard all triples $(w_h, w_{...}, w_t)$
    end if
  end for
end for ▷ The sets are now grouped by (w_h, w_t)
Merge all translation sets containing triples with same (w_h, w_m)
Merge all translation sets containing triples with same (w_m, w_t)

presented in Figure 3. The highest $\text{score}(w_h, w_t)$ is 0.143. When $\alpha = 0.8$ and $\beta = 0.2$, (w_h, w_t) pairs whose score is less then $\alpha \times 0.143 = 0.1144$, or whose score squared is less then $\beta \times 0.143 = 0.0286$ will be discarded. Therefore, triples containing (garang, 大胆) (and other pairs of lower scores) will be discarded as its score 0.111 and squared score 0.0123 are lower than both threshold values.

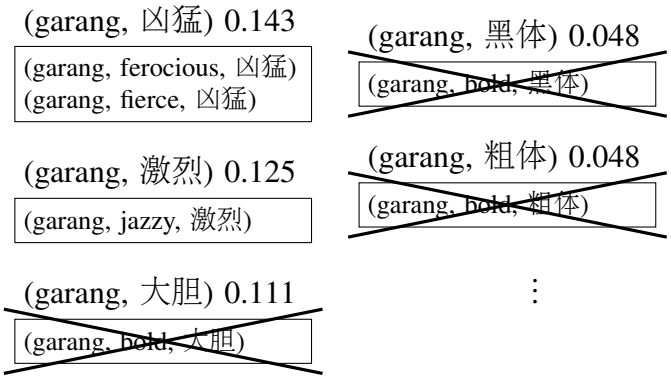


Fig. 3. Generated translation triples from Algorithm 1

The retained translation triples are then merged into translation sets based on overlapping translation pairs among the languages. An example is shown in Figure 4, where the translation triples are merged into one translation set with five members.

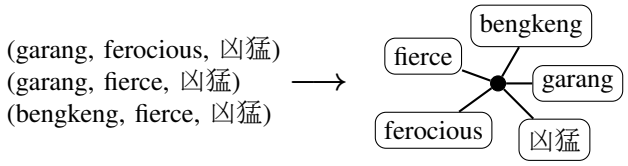


Fig. 4. Merging translation triples into translation sets

C. Adding More Languages

The algorithm described in the previous section gives us a trilingual translation lexicon for languages $\{L_1, L_2, L_3\}$. Algorithm 2 outlines how a new language L_4, or more generally, L_{k+1} can be added to an existing multilingual lexicon of languages $\{L_1, L_2, \ldots, L_k\}$. We first run OTIC to produce translation triples for L_{k+1} and two other languages already included in the existing lexicon. These new triples are then compared against the existing multilingual entries. If two words in a triple are present in an existing entry, the third word is added to that entry as well.

Figure 5 gives such an example: given the English–Chinese–Malay translation set earlier, we prepare translation triples for French–English–Malay. By detecting overlapping English–Malay translation pairs in the translation set and

Algorithm 2: Adding L_{k+1} to multilingual lexicon $\mathbb{L}$ of $\{L_1, L_2, \ldots, L_k\}$

1: $T \leftarrow$ translation triples of L_{k+1}, L_m, L_n generated by Algorithm 1 where $L_m, L_n \in \{L_1, L_2, \ldots, L_k\}$
2: **for all** $(w_{L_m}, w_{L_n}, w_{L_{k+1}}) \in T$ **do**
3: Add $w_{L_{k+1}}$ to all entries in $\mathbb{L}$ that contains both w_{L_m} and w_{L_n}
4: **end for**

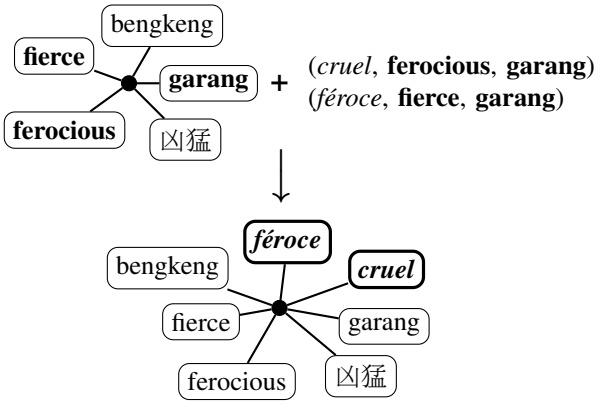


Fig. 5. Adding French members to existing translation sets

triples, two new French words '*cruel*' and '*féroce*' are added to the existing translation set.

D. Resources for Experiment

We generated a multilingual lexicon for Malay, English and Chinese using the modified OTIC procedure, with English as the intermediate language. We used the following bilingual dictionaries as input:

- *Kamus Inggeris–Melayu untuk Penterjemah*, an English to Malay dictionary published by PTS Professional Publishing. The vast majority of Malay glosses in this dictionary are single words, or simple phrases containing only a few words. We therefore reversed the direction and used it as a Malay to English dictionary.
- *XDict*, a free English to Chinese Dictionary packaged for GNU/Linux distros, including Ubuntu and Debian.
- *CC-CEDICT*[1], a free Chinese to English dictionary. We omitted Chinese lexical items marked to be archaic, idioms and family names. As CC-CEDICT entries do not include a part-of-speech (POS) field, we assigned one to each entry–gloss pair by running the Stanford POS Tagger[2] on the English glosses.

We normalised English entries with respect to American and British spelling variances[3], as well as stripping off the verb infinitive '*to*'. Chinese entries were normalised by stripping off the adjective marker '的'. (See [9] for other normalisation possibilities.)

[1]http://cc-cedict.org/wiki/start
[2]http://nlp.stanford.edu/software/tagger.shtml
[3]http://wordlist.sourceforge.net/

To add French to the generated Malay–English–Chinese lexicon, we converted entries from FeM, a French–English–Malay dictionary[4], into translation triples with default scores of 1.0.

We provided a look-up interface to the resultant multilingual lexicon, using which users can look up a word in any member languages. All multilingual entries containing the word being looked up will be returned, with the words inside each entry being ranked by their associated OTIC scores. Figure 6 shows the look-up results for Malay '*kebun*'.

7566

English	Bahasa Malaysia	中文	français
farm (0.38);	kebun (0.42); ladang (0.34);	农场 (0.45); 饲养场 (0.31);	fermé (1.00); fermier (1.00);

8623

English	Bahasa Malaysia	中文	français
garden (0.42);	kebun (0.50); taman (0.33);	花园 (0.42);	jardin (1.00);

Fig. 6. Multilingual lexicon look-up result for Malay '*kebun*'

IV. Results and Discussion

As the correct addition of French lexical items depends on the accuracy of the Malay–English–Chinese lexicon generated, and also because it was harder for us to find evaluators who speak all four languages, only the Malay–English–Chinese entries are evaluated.

In general, precision increases for greater threshold values of α and β, at the expense of less words in each language being included. Our procedure produced more translation sets which should have been merged (false negatives) when α and β are high; however this is more desirable than words of different meanings being placed in the same translation set (false positives).

We performed two evaluations on the generated multilingual lexicon, described in the following subsections.

A. Evaluation on 100 Random Translation Sets

For the first evaluation, we randomly selected 100 translation sets constructed from at least two translation triples, using different α and β values. Evaluators were told to only accept as accurate translation sets in which all member Malay, English and Chinese words are (near-)synonyms. For this initial work, a translation set is deemed accurate if it contains

[4]http://www-clips.imag.fr/cgi-bin/geta/fem/fem.pl?lang=en

diversified word meanings, i.e. it is acceptable for both Malay '*beras*' (uncooked rice) and '*nasi*' (cooked rice) to occur in the same translation set as English '*rice*'. The evaluation results are summarised in Figure 7.

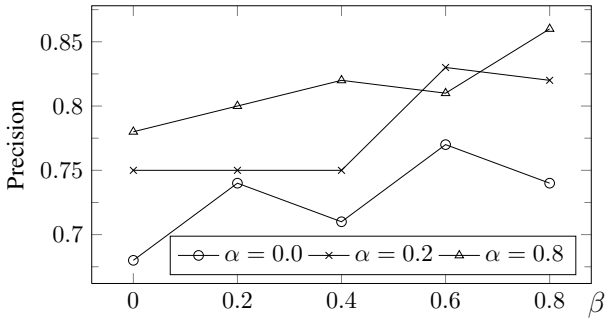


Fig. 7. Precision for 100 randomly selected translation sets with varying α and β.

Precision increases with α and β, but is generally in the range of 0.70–0.82, and can go up to as high as 0.86. Of the erroneous sets, most of the wrongly included words are not the top-ranked ones in each language, especially when α and β are high. Many errors are caused by incorrect POS assignments to *CEDICT* entries. Nevertheless, we find such results encouraging, particularly because it can be achieved with such simple bilingual translation mapping lists.

B. Evaluation on Test Word Samples

As mentioned earlier near the end of section III-B, translation sets generated using OTIC are most prone to error when the intermediate language (English in our experiment) word is polysemous, thereby selecting a 'tail' language word that does not have the same meaning as the 'head' language word.

To evaluate how effective OTIC is at detecting polysemy in the intermediate language, we selected four polysemous English words as test words, namely '*bank*', '*plant*', '*target*' and '*letter*'. We define a list of gold standard translation sets for each test word, based on all possible generated triples from our input dictionaries. All translation sets containing the test words are then retrieved. By viewing generation of translation sets as a data clustering problem, we access their accuracy by calculating the F_1 score and Rand index (RI) [11] for each list of retrieved translation sets $R = \{R_1, R_2, \ldots, R_m\}$ for a test word against that test word's golden standard $A = \{A_1, A_2, \ldots, A_n\}$:

$\text{TP} = |\{\text{word pairs occurring in some } R_i \in R \text{ and some } A_j \in A\}|$
$\text{FP} = |\{\text{word pairs occurring in some } R_i \in R \text{ but not in any } A_j \in A\}|$
$\text{TN} = |\{\text{word pairs not occurring in any } R_i \in R \text{ nor any } A_j \in A\}|$
$\text{FN} = |\{\text{word pairs not occurring in any } R_i \in R \text{ but in some } A_j \in A\}|$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{RI} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Table I, figures 8 and 9 summarise the results for each test word. Here again, RI and F_1 increase with α and β. We also note from the graphs that the threshold β is more influential in raising both RI and F_1. However, the scores may decrease if α and β are too high, as is the case for '*plant*' when $\alpha = 0.8$ and $\beta \geq 0.4$. This is due to valid words being rejected by the high thresholds, thereby increasing the number of false negatives (FN) and lowering RI and F_1. Taking into account the Rand indices, F_1 scores as well as word coverage in each language, we found taking $\alpha \approx 0.6$ and $\beta \approx 0.2$ to offer a reasonable balance between precision, recall and coverage.

TABLE I
MINIMUM AND MAXIMUM RAND INDEX AND F_1 SCORE FOR EACH TEST WORD

Test word	Rand Index min	Rand Index max	F_1 min	F_1 max	Min. thresholds for best score α	Min. thresholds for best score β
'*bank*'	0.417	0.611	0.588	0.632	0.6	0.4
'*plant*'	0.818	0.927	0.809	0.913	0.6	0.2
'*target*'	0.821	1.000	0.902	1.000	0.4	0.2
'*letter*'	0.709	0.818	0.724	0.792	0.8	0.2

V. RELATED WORK

There have been many efforts to create lexical databases similar to the Princeton English WordNet [5] for other languages. To leverage the many types of rich data and resources built on top of Princeton WordNet, many such projects aim to align their entries to those in the Princeton WordNet. Notable wordnet projects include EuroWordNet (Western European languages) [3], BalkaNet (Eastern European languages) [2], and many more[5]. All these wordnets taken together can be regarded as a huge multilingual lexicon, with the Princeton Wordnet as its main hub. However, this also means these wordnets tend to suffer from a frequent critique against the Princeton WordNet: its overly fine sense distinctions often cause human lexicographers and evaluators working with the wordnets much confusion, as well as complicating NLP applications that make use of them.

Sammer and Soderland [4] constructed PanLexicon, a multilingual lexicon by computing context vectors for words of different languages from monolingual corpora, then grouping the words into translation sets by matching their

[5]see http://www.globalwordnet.org/gwa/wordnet_table.htm

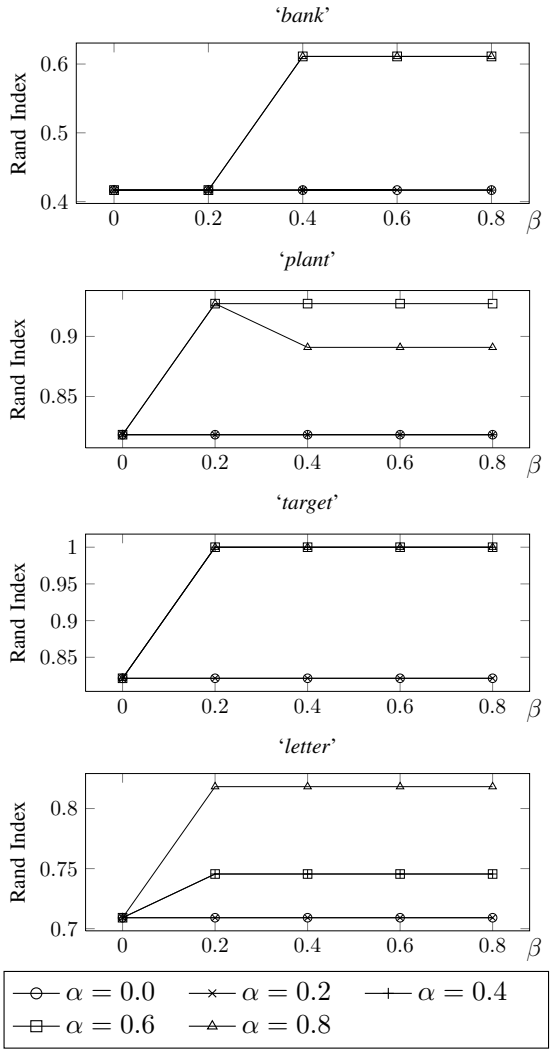


Fig. 8. Rand indices for translation sets containing '*bank*', '*plant*', '*target*' and '*letter*' with varying thresholds α and β.

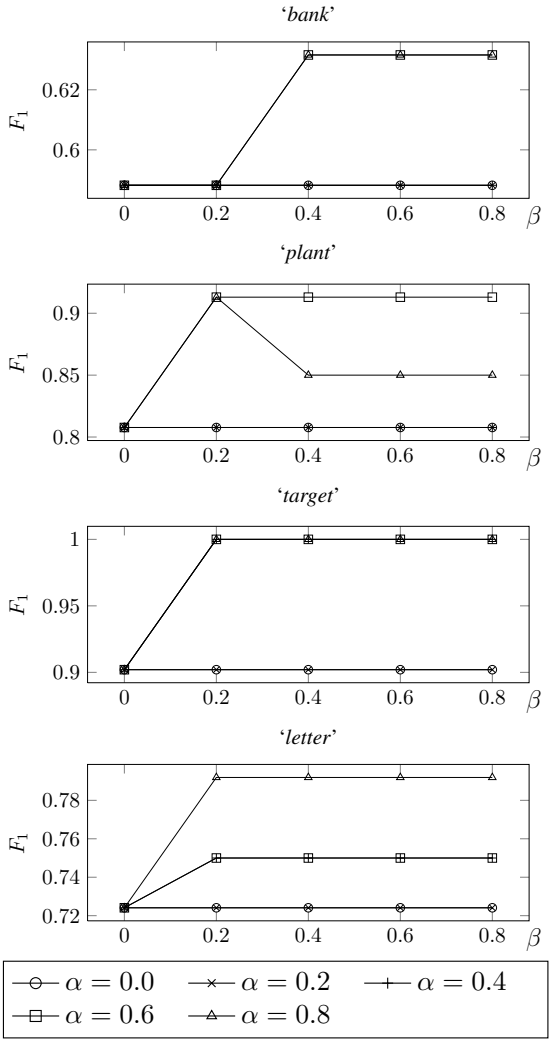


Fig. 9. F_1 scores for translation sets containing '*bank*', '*plant*', '*target*' and '*letter*' with varying thresholds α and β.

context vectors with the help of bilingual lexicons. By using a corpus-based method, good coverage of words from different languages is expected. In addition, sense distinctions are derived from corpus evidence, which are unlikely to be as fine as those of Princeton WordNet. However, their method produces many translation sets that contain semantically related but not synonymous words, e.g. '*shoot*' and '*bullet*', thus lowering the precision: the authors report 44 % precision based on evaluators' opinions (75 % if inter-evaluator agreement is not required). In addition, specific methods for identifying multi-word expressions (MWEs) in the corpus are required (which was not taken into consideration in their paper), whereas our method would also process MWEs if they are listed in the bilingual lexicons.

Markó, Schulz and Hahn [12] made use of cognate mappings to derive new translation pairs, later validated by processing parallel corpora in the medical domain. Due to the special characteristics of medical terms, each complex term is indexed on the level of sub-words, e.g. '*pseudo⊕hypo⊕para⊕thyroid⊕ism*'. The authors report up to 46 % accuracy for each language pair by checking against data from the Unified Medical Language System (UMLS). The biggest drawback in their approach is the requirement for large aligned thesaurus corpora, although such resources may be more readily available for specific domains such as medicine. Also, the cognate-based approach would not be applicable for language pairs that are not closely related.

Lafourcade [1] also uses a vector-based model for populating the Papillon multilingual dictionary [7]. Instead of constructing context vectors from corpora, Lafourcade computes conceptual vectors for each translation pair from a bilingual dictionary, based on the gloss text (written in the source language) and associated class labels from a semantic hierarchy. Translation pairs of different language pairs are then compared based on their conceptual vectors to determine if they express the same meaning. By using class labels as

the vector space generator, the conceptual vector model is able to merge dictionary entries whose gloss text contain synonymous words. It does, however, require the class labels to be assigned to the dictionary entries. Such resources are not always available, and the additional task of assigning class labels is time-consuming and costly.

VI. Conclusion and Future Work

We have described a low cost procedure for constructing a multilingual lexicon using only simple bilingual translation lists, suitable especially for including under-resourced languages in lexical resources. Precision of random samples of the generated translation sets averages in the range of 0.70–0.82. Based on the experimental Rand indices and F_1 scores for selected lexical samples, we found threshold values of $\alpha \approx 0.6$ and $\beta \approx 0.2$ give reasonable balance between precision, recall and word coverage.

Manually validating and correcting an automatically constructed lexicon, entry by entry, can be very costly both in time and human expertise. We plan to take another approach, by deploying the bootstrapped multilingual lexicon in a machine translation system and capturing user actions when they edit the translation to update the lexicon entries.

Acknowledgment

The work reported in this paper is supported by a Fundamental Research Grant (FRGS/1/10/TK/MMU/02/02) from the Malaysian Ministry of Higher Education. We thank the evaluators who participated in the results evaluation, and the two anonymous reviewers for their comments on improving this paper.

References

[1] M. Lafourcade, "Automatically populating acception lexical database through bilingual dictionaries and conceptual vectors," in *Proceedings of PAPILLON-2002*, Tokyo, Japan, 8 2002.

[2] D. Tufiş, D. Cristeau, and S. Stamou, "BalkaNet: Aims, methods, results and perspectives – a general overview," *Romanian Journal of Information Science and Technology Special Issue*, vol. 7, no. 1, pp. 9–43, 2004.

[3] P. Vossen, "EuroWordNet: A multilingual database of autonomous and language-specific wordnets connected via an Inter-Lingual-Index," *Special Issue on Multilingual Databases, International Journal of Linguistics*, vol. 17, no. 2, 2004.

[4] M. Sammer and S. Soderland, "Building a sense-distinguished multilingual lexicon from monolingual corpora and bilingual lexicons," in *Proceedings of Machine Translation Summit XI*, Copenhagen, Denmark, 2007, pp. 399–406.

[5] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*, ser. Language, Speech, and Communication. Cambridge, Massachusetts: MIT Press, 1998.

[6] G. Francopoulo, N. Bel, M. George, N. Calzolari, M. Monachini, M. Pet, and C. Soria, "Multilingual resources for NLP in the lexical markup framework (LMF)," *Language Resources and Evaluation*, vol. 43, no. 1, pp. 57–70, 3 2009.

[7] C. Boitet, M. Mangeot, and G. Sérasset, "The PAPILLON project: Cooperatively building a multilingual lexical database to derive open source dictionaries & lexicons," in *Proceedings of the 2nd Workshop on NLP and XML (NLPXML'02)*, 2002, pp. 1–3.

[8] K. Tanaka, K. Umemura, and H. Iwasaki, "Construction of a bilingual dictionary intermediated by a third language," *Transactions of the Information Processing Society of Japan*, vol. 39, no. 6, pp. 1915–1924, 1998, in Japanese.

[9] F. Bond and K. Ogura, "Combining linguistic resources to create a machine-tractable Japanese–Malay dictionary," *Language Resources and Evaluation*, vol. 42, pp. 127–136, 2008.

[10] F. Bond, b. S. Ruhaida, T. Yamazaki, and K. Ogura, "Design and construction of a machine-tractable Japanese-Malay dictionary." in *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain, 2001, pp. 53–58.

[11] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[12] K. Markó, S. Schulz, and U. Hahn, "Multilingual lexical acquisition by bootstrapping cognate seed lexicons," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP) 2005*, Borovets, Bulgaria, 2005.

A Cross-Lingual Pattern Retrieval Framework

Mei-Hua Chen, Chung-Chi Huang, Shih-Ting Huang, Hsien-Chin Liou, and Jason S. Chang

***Abstract*—We introduce a method for learning to grammatically categorize and organize the contexts of a given query. In our approach, grammatical descriptions, from *general* word groups to *specific* lexical phrases, are imposed on the query's contexts aimed at accelerating lexicographers' and language learners' navigation through and *GRASP* upon the word usages. The method involves lemmatizing, part-of-speech tagging and shallowly parsing a general corpus and constructing its inverted files for monolingual queries, and word-aligning parallel texts and extracting and pruning translation equivalents for cross-lingual ones. At run-time, grammar-like patterns are generated, organized to form a thesaurus index structure on query words' contexts, and presented to users along with their instantiations. Experimental results show that the extracted predominant patterns resemble phrases in grammar books and that the abstract-to-concrete context hierarchy of querying words effectively assists the process of language learning, especially in sentence translation or composition.**

***Index terms*—Grammatical constructions, lexical phrases, context, language learning, inverted files, phrase pairs, cross-lingual pattern retrieval.**

I. Introduction

MANY language learners' queries (e.g., "play" or "role") are submitted to computer-assisted language learning tools on the Web for word definitions or usages every day. And an increasing number of Web services specifically target English as Foreign Language (EFL) learners' search questions.

Web-based language learning tools such as Sketch Engine, concordancers, and TANGO typically take monolingual single-word query and retrieve too many its collocations and example sentences such that they overwhelm and confuse users due to the amount of returned sentences and different usages therein. However, users may want to learn the context patterns, or grammatical sequences underlying contextual word strings, (e.g., 'play article adjective role') of a specific word sense of a word and submit multiple-word queries (e.g., "play role"), and users may need an index to quickly navigate through one usage to another. Besides, EFL users may prefer submitting queries in their first languages. These queries could be answered more appropriately if a tool provided grammatical categories to their contexts and understood other languages.

Manuscript received November 28, 2010. Manuscript accepted for publication January 5, 2011.

Mei-Hua Chen, Chung-Chi Huang, Shih-Ting Huang, and Jason S. Chang are with ISA, NTHU, HsinChu, Taiwan, R.O.C. 300 (e-mail: {chen.meihua, u901571,koromiko1104,Jason.jschang}@gmail.com).

Hsien-Chin Liou is with FL, NTHU, HsinChu, Taiwan, R.O.C. 300 (e-mail: liuhsienc@gmail.com).

Consider the learner query "play role". The best response is probably not the overwhelming set of sentences containing "play role". A good response might generalize and categorize its representative contexts such as: "play role" separated by "DT JJ" (common instantiation: "an important") where "DT" denotes an article and "JJ" an adjective, "play role" followed by "IN VBG" (instantiation: "in determining") where "IN" denotes a preposition and "VBG" a gerund, and "play role" preceded by "NN MD" (instantiation: "communication will") where "NN" denotes a noun and "MD" an auxiliary verb. Such generalization and categorization of the query's contexts can be achieved by part-of-speech (PoS) tagging its sentences. Intuitively, by word-class or PoS information, we can bias a retrieval system towards grammar-like pattern finder. On the other hand, by leveraging machine translation techniques, we can channel the first-language query to its English substitutes.

We present a new system, *GRASP* (grammar- and syntax-based pattern-finder) that automatically characterizes the contexts of querying collocations or phrases in a grammatical manner. An example cross-lingual *GRASP* search for the Chinese collocation "扮演角色" ("play role" or "play part") is shown in Figure 1. *GRASP* has directed the first-language query "扮演角色" to one of its probable English translations, "play role", and gathered its predominant patterns of phraseology in terms of the relative position between the query and its contexts, and the distances between the querying words, based on a balanced monolingual corpus. Take the most frequent distance (i.e., 3) where "play" and "role" are apart from each other for example. "Play" and "role" are most likely to be separated by word group "DT JJ", constituting the lexically open formal idiom or grammatical construction "play DT JJ role" what we call *GRASP* syntactic pattern. And this *GRASP* pattern's frequent idiomatic lexical realizations or phrases, or lexically filled substantive idioms[1], are "play an important role". To extract such formal or substantive idioms, *GRASP* learns translations and word-to-sentence mappings automatically (Section 3).

At run-time, *GRASP* starts with an English query or a first-language query for usage learning.

GRASP then retrieves aforementioned formal idioms lexically anchored with English query words' lemmas and their substantive counterparts/instantiations. The former are designed for quick word usage navigation and the latter for better understanding of phraseological tendencies.

[1] See (Fillmore *et al.*, 1988).

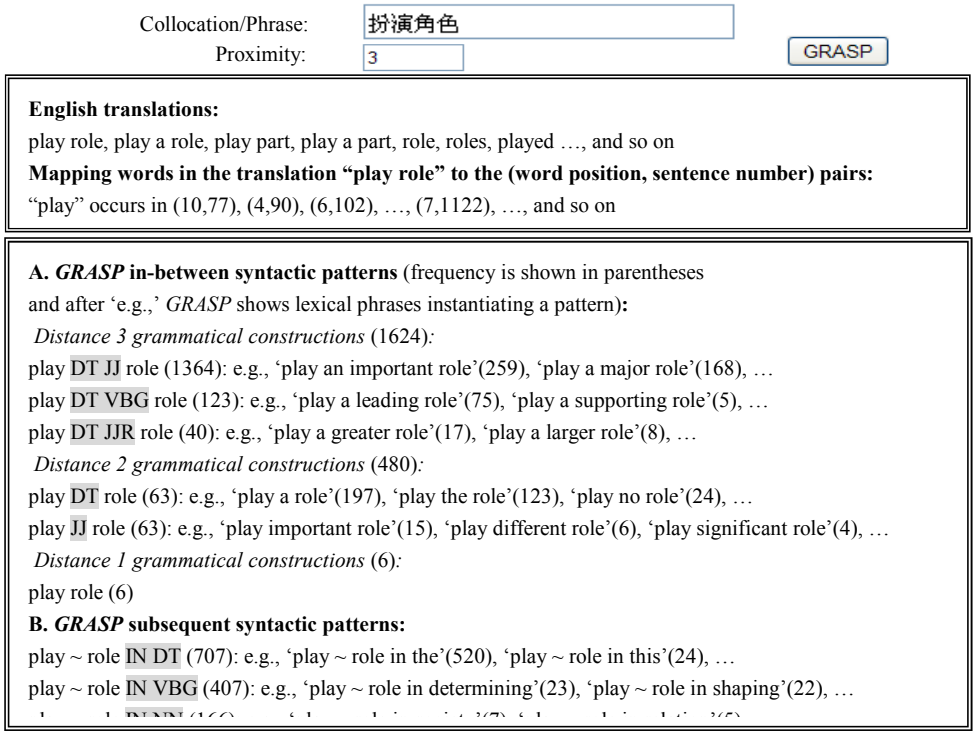


Fig. 1. An example *GRASP* response to the query "扮演角色" ("play role").

In our prototype, *GRASP* accepts queries of any length and responds with example sentences and frequencies of the formal or substantive idioms.

II. Related Work

Ever since large-sized corpora and computer technology became available, many linguistic phenomena have been statistically modeled and analyzed. Among them is collocations long been considered essential in language learning. In the beginning, collocations are manually exemplified and examined (Firth, 1957; Benson, 1985; Benson *et al.*, 1986; Sinclair 1987; Lewis, 2000; Nation, 2001). Right after a pioneering statistical analysis on collocations (Smadja, 1993), the area of research soon becomes computationally possible (Kita and Ogata, 1997) and active especially in English for academic purpose (Durrant, 2009) or second language learning (e.g., (Liu, 2002) and (Chang *et al.*, 2008)).

Recently, some collocation finders such as *Sketch Engine*, *TANGO* and *JustTheWord* have been developed and publicly available. *Sketch Engine* (Kilgarriff *et al.*, 2004) summarizes a word's collocational behavior. *TANGO* (Jian *et al.*, 2004) further provides cross-language searches while *JustTheWord* automatically clusters co-occurring words of queries. In this paper, we take note of the regularities of words' contexts and grammatically express the regularities as patterns for language learning. Such patterns go beyond the collocations from collocation finders, possibly limited to certain combinations of lexical or grammatical collocations and missing the important contextual word groups or words of the collocations.

Textual cohesion is observed in phrases as well. Therefore, phraseology and pattern grammar have drawn much attention. Phraseology can be studied via lexically fixed word sequences, i.e., n-grams (Stubbs, 2002), or totally lexical-open PoS-grams (Feldman *et al.*, 2009; Gamon *et al.*, 2009). In contrast to these two extremes, Stubbs (2004) introduces phrase-frames (p-frames) which bases on n-gram but with one variable slot. Our framework lies between n-grams and PoS-grams, and our extracted sequences of patterns consist of more-than-one variable slots featuring the contexts surrounding the querying words.

Recent work has been done on statistically analyzing the contexts, patterns, frames or constructions of words. A lexical-grammatical knowledge database, StringNet, is built and described in (Wible *et al.*, 2010). However, their work may over-generalize the querying words to word groups during database construction and does not handle multi-word or cross-lingual queries. Users of language tools may submit these two types of queries in that patterns are closely associated with meanings, senses of words, and multiple words usually restrains the senses of words (see (Yarowsky, 1995)), and users may experience problems composing queries in the language they are learning. Hence, we propose a multi-word and cross-lingual pattern-retrieval framework in which patterns are anchored with users' querying words with their contextual words generalized. In a study more related to our work, (Cheng *et al.*, 2006) describes the concept of conc-grams and how to use conc-grams to find constituency and positional variants of search words. The main difference from their work is that we give descriptions to query words' predominant contexts in a grammatical and systematic manner. The descriptions are thesaurus index structures, consisting of

constructions from lexically-open syntactic patterns to lexically fixed idioms.

III. THE *GRASP* FRAMEWORK

A. Problem Statement

We focus on imposing a thesaurus index structure on the querying words' contexts. This structure, formed by a hierarchy from *general* (lexically open) grammatical constructions to *specific* (lexically fixed) substantive idioms anchored with query words, provides a means for quick navigation and understanding of words' typical patterns and their instantiated lexical phrases, and is returned as the output of the system. The returned constructions, or patterns can be examined by learners and lexicographers directly or a syntax-based machine translation system. Thus, it is crucial that the set of patterns cannot be so large that it overwhelms the user. At the same time, there is a need for first-language query search among EFL learners. Therefore, our goal is to return a reasonable-sized set of recurrent grammatical patterns and their idiomatic lexical realizations for language learning or lexicography that represents queries' attendant phraseology and expected lexical items, taking both monolingual and cross-lingual query search. We now formally state the problem that we are addressing.

Problem Statement: We are given a large-scale general corpus *C* (e.g., British National Corpus), a parallel text *T* (e.g., Hong Kong Parallel Text), and a query phrase *Q*. Our goal is to extract and organize the contexts of the query *Q* lexico-grammatically and lexically based on *C* that are likely to assist users in navigating and learning the usages of *Q*. For this, we transform words $w_1,\ldots, w_m$ in *Q* into sets of (word position, sentence record) pairs such that the top *N* lexico-grammatical patterns and their lexical instances depicting the query's context are likely to be quickly retrieved. *T*, on the other hand, makes cross-lingual query and learning possible.

B. Corpora Preprocessing

In the corpora preprocessing, we attempt to find transformations from words in the query into (position, sentence) pairs, collocations for single-word query for starters, and English translations for first-language query, expected to accelerate the search for *GRASP* grammatical patterns and expected to accommodate EFL learners' habits of composing a query.

Lemmatizing, PoS Tagging and Shallow Parsing. In the first stage of the preprocessing, we lemmatize each sentence in the general corpus *C* and generate its most probable PoS tag sequence and shallow parsing result. The goal of lemmatization is to reduce the impact of inflectional morphology of words on statistical analyses while that of PoS tagging is to provide a way to grammatically describe and generalize the contexts/usages of a collocation/phrase. Shallow parsing results, on the other hand, provide the base phrases of a sentence. And consecutive base phrases are often used for extracting collocation candidates.

Finding Collocations. In the second stage of the preprocessing process, we identify a set of reliable collocations in *C* based on statistical analyses. Collocations of single-word queries may be presented to language learners with, to some extent, few clues, as starters for more complete and specific queries.

The input to this stage is a set of lemmatized, PoS tagged and shallowly parsed sentences while the output of this stage is a set of statistically-suggested collocations. The method for finding reliable collocations in *C* consists of a number of steps, namely, determining the head words in the base phrases from shallow parser, constituting the head words as collocation candidates, calculating the pair-wise mutual information (MI) values of the head words, and filtering out the collocation candidates whose MI values do not exceed an empirical threshold.

Considering the enrichment (usually adjectives and prepositions) *GRASP* can offer and the observation that EFL learners have hard time composing sentences with verb-noun (VN) collocations and choosing right following prepositions, collocation type to bridge single-word query focuses on VN and verb-preposition (VP) collocation. Focusing on VN collocations and VP collocations, we highlight the contiguous verb phrase and noun phrase, and verb phrase and prepositional phrase in *C*. In the highlighted verb, noun and prepositional phrases, we intuitively consider their last verb, noun and preposition to be the head words and constitute collocation candidate of the form $<\text{word}_1,\text{pos}_1,\text{word}_2,\text{pos}_2>$ based on the two head words in the two base phrases. To examine the candidates, we compute MI values using

$$\text{MI}=\log(\mathit{freq}(\text{word}_1,\text{pos}_1,\text{word}_2,\text{pos}_2)/(\mathit{freq}(\text{word}_1,\text{pos}_1)\times\mathit{freq}(\text{word}_2,\text{pos}_2)))$$

in which *freq*(*) denotes the frequency. MI values have been used to determine the mutual dependence of two events. The higher the MI values, the more dependent they are. At last, we retain only candidates whose MI values exceed threshold Θ and think of them as statistically-suggested collocations.

Constructing Inverted Files. In the third stage of preprocessing, we build up inverted files for the lemmas in the corpus *C*. For each lemma in *C*, we record the positions and sentences in which it resides for run-time query. Additionally, its corresponding surface word form, PoS tag and shallow parsing result are kept for reference in that such information gathered across lemmas is useful in grammatical pattern finding and (potentially) language learning.

Word-aligning and phrase pairs extracting. In the fourth stage, we exploit a large-scale parallel text *T* for bilingual phrase acquisition, rather than using a manually compiled dictionary to achieve satisfying translation coverage and variety.

We acquire phrase pairs via the following procedure. First, we word-align the bitext in *T* leveraging the IBM model 1 to model 5 implemented in *GIZA++* (Och and Ney, 2003). To "smooth" the saw-toothed word alignments produced by directional word alignment model of IBM and collect words with no translation equivalent in another language in phrases,

grow-diagonal-final is used for bidirectional word alignment combination. Finally, heuristics in (Koehn *et al.*, 2003) are used for bilingual phrase extracting.

Pruning unlikely phrase pairs. In the fifth and final stage of the preprocessing, we filter out less probable or insignificant translation equivalents obtained from *T*. In this paper, we apply the pruning techniques described in (Johnson *et al.*, 2007). Specifically, we use their significance testing of phrases to first prune insignificant phrase pairs and rank the English translations of the first-language search queries. For language learning, an accurate and small but diverse set of translations are especially helpful. Moreover, *GRASP* patterns will be shown for the translations, if triggered or automatically, which further provides the hierarchical index for navigation through specific usages and word associations in English for the query initially in users' mother tongue. One thing worth mentioning is that the set of translation equivalents outputted in this stage includes those in which we skip some word pairs in the phrase pairs, in order to increase the translation coverage for the first-language queries. The skipped phrase pairs are constructed as follows. For each phrase pair, we skip some number of the words on the first-language end and if the skipped words have word alignments on the English part, the aligned English words are also skipped. Then we constitute the un-skipped words in the two languages as a skipped phrase pair.

C. Run-Time Index Structure Building and Pattern Finding

Once collocates, word-to-sentence mappings, and confident phrase pairs are obtained, *GRASP* constructs the thesaurus index hierarchy for English contexts and phraseology of the query using the procedure in Figure 2.

In Step (1) of the algorithm we reformulate the user-nominated query into a set of new queries, *Queries*, if necessary. The first type of the reformulation concerns the language used for the input *query*. If *query* is in a language other than that of *C*, we translate the *query* into its statistically significant (English) translations based on the pruned and skipped phrase tables from *T*, and append each of these translations to *Queries* considering it as a search query as if it were submitted by the user. The second concerns the length of the query. Since presenting single word alone to *GRASP* is uncertain with its word sense in question and contexts or pattern grammars are typically highly associated with a word's meanings, for single-word queries, we use their reliable collocations, specifically VN and VP ones, obtained from Section 3.2 as stepping stones to *GRASP* syntactic patterns. These again are incorporated into *Queries*. Note that for these two kinds of query transformation, users may be allowed to choose their own interested translation or collocation of the *query* in implementation and presented only with its (i.e., the translation's or collocation's) *GRASP* hierarchy of word usages. The prototypes for first-language, Chinese in particular, queries and monolingual single-word or multi-word queries are at http://140.114.214.80/theSite/*GRASP*_v552/ and http://140.114.214.80/theSite/b*GRASP*_v552/ respectively. In Step (2) we initialize a set *GRASPresponses* to collect *GRASP* grammatical patterns of queries in *Queries* now in English and more-than-one words.

```
procedure GRASPindexBuilding(query,proximity,N,C,T)
(1)  Queries=queryReformulation(query)
(2)  GRASPresponses= ϕ
     for each query in Queries
(3)    interInvList=findInvertedFile(w1 in query)
       for each lemma wi in query except for w1
(4)      InvList=findInvertedFile(wi)
         //perform AND operation on interInvList and InvList
(5a)     newInterInvList= ϕ ; i=1; j=1
(5b)     while i<=length(interInvList) and j<=lengh(InvList)
(5c)       if interInvList[i].SentNo==InvList[j].SentNo
(5d)         if withinProximity(interInvList[i].wordPosi,
                 InvList[j].wordPosi, proximity)
(5e)           Insert(newInterInvList, interInvList[i],InvList[j])
             else if interInvList[i].wordPosi<InvList[j].wordPosi
(5f)           i++
             else //interInvList[i].wordPosi>InvList[j].wordPosi
(5g)           j++
           else if interInvList[i].SentNo<InvList[j].SentNo
(5h)         i++
           else //interInvList[i].SentNo>InvList[j].SentNo
(5i)         j++
(5j)     interInvList=newInterInvList
     //GRASP thesaurus index building
(6)  PatternIndex= ϕ  // a collection of patterns for this query
     for each element in interInvList
(7)    PatternIndex+= {GrammarPatternGeneration(query,element,C)}
(8a) Sort patterns and their instances in PatternIndex
     in descending order of frequency
(8b) GRASPresponse=top N patterns and instances in PatternIndex
(9)  append GRASPresponse to GRASPresponses
(10) return GRASPresponses
```

Fig. 2. Run-Time Index Building and Pattern Finding.

In Step (3) *interInvList* is initialized to contain the intersected inverted files of the lemmas in the *query*. For each lemma w_i in *query*, we obtain its inverted file, *InvList* (Step (4)) before performing an AND/intersection operation on *interInvList*, intersected results from previous iteration, and *InvList* (from Step (5a) to (5j)[2]). The AND operation is defined as follows. First, we enumerate the inverted lists, *interInvList* and *InvList* (Step (5b)) after the initialization of their respective indices (i.e., *i* and *j*) and temporary resulting list *newInterInvList* (Step (5a)). Second, we incorporate a new instance into *newInterInvList* (Step (5e)) if the sentence records of the indexed elements of *interInvList* and *InvList* in question are the same (Step (5c)) and the distance between the word positions of these elements are within *proximity* (Step (5d)). Note that, in Step (5e), a new instance of (word position, sentence record) is created based on *interInvList*[*i*] and *InvList*[*j*] and inserted into *newInterInvList*. Furthermore, taking into account the positional variations of a

[2] These steps only hold for sorted inverted files.

collocation/phrase (e.g., "play role" and "role play"), function withinProximity of Step (5d) considers the *absolute* difference between word positions, to cover contexts of differently-ordered querying words. Finally, we set *interInvList* to be *newInterInvList* for the next iteration of the AND operation (Step (5j)).

After finding the legitimate sentences containing a query's words within certain distance, *GRASP* retrieves and builds the hierarchical index structure for its contexts. In Step (7) we generate grammar patterns or cases of word usages for each *element*, taking the form ([wordPosi(w_1), …, wordPosi(w_i), …], sentence number) pointing out the validated sentence record and the word positions of the query's lemmas in that sentence, in *interInvList*. In function GrammarPatternGeneration, based on *element* and *C*'s lemmas and PoS tags, we first transform the legitimate sentence by replacing its words with PoS tags except for the words in positions [wordPosi(w_1), …, wordPosi(w_i), …] and replacing these words with lemmas. Afterwards, we extract contiguous segments surrounding the query lemmas from the transformed sentence, resulting in syntax-based context of the search query (e.g., "play DT JJ role" and "play ~ role IN VBG"). Such lexically open pattern grammars representing the regularity of words' contexts are referred to as *GRASP* syntactic patterns in this paper. Very similarly, the lexically fixed realizations of these patterns could be extracted.

We collect the *N* most frequent (recurrent or potentially idiomatic) *GRASP* syntactic patterns and their *N* most frequent realizations (Step (8)), and gather them as a *GRASP* response *GRASPresponse*. At last, we return all the responses (i.e., *GRASPresponses*) that may interest our users. Figure 1 illustrates the summarized grammatical context ontology for "play role" from a Chinese query "扮演角色".

D. Further Improvement to GRASP

In this subsection, we manage to further extend the *GRASP* patterns. The extension is made in two ways: lexicalization and sub-categorization.

TABLE I
PATTERNS BEFORE AND AFTER LEXICALIZATION

Query	Before	After
play role	play ~ role IN DT (707)	play ~ role IN(in) DT (599)
	play ~ role IN VBG (407)	play ~ role IN(in) VBG (397)
	role ~ play IN DT (235)	role ~ play IN(in) DT (128)
		role ~ play IN(by) DT (89)
have effect	have ~ effect IN DT (1199)	have ~ effect IN(on) DT (887)
	have ~ effect IN VBG (644)	have ~ effect IN(of) VBG (533)
		have ~ effect IN(upon) DT (83)

In writing we observe that EFL learners often have difficulty choosing the right preposition following a collocation (e.g., VN, AN, and PN collocation). Therefore, we lexicalize on the IN PoS tag, a prepositional PoS tag, in *GRASP* patterns to present the specific prepositions to users. Table I shows example *GRASP* patterns before and after lexicalization. Note that lexicalization is indicated in parentheses and that the statistics of frequencies (numbers in parentheses) may change.

Secondly, to acquire grammar rules such as "provide SOMEBODY with SOMETHING" and "provide SOMETHING to SOMEBODY" in grammar books, we semantically subcategorize PoS tags in *GRASP* patterns. Although some current patterns may be informative enough in terms of the semantic roles of the PoS tags, some are not especially the ones with the too general PoS tags NN and NNS, standing for singular and plural nouns respectively. We thus classify the semantic roles of these tags in *GRASP* patterns.

We now describe our simple strategy for semantic role categorization, relying on a lexical thesaurus with words' semantic roles or meanings. In our implementation, we use WordNet where each sense of a word has a higher-level and more abstract supersense, or lexicographers' file. The strategy first, for each extracted pattern accompanied with words of the NN and NNS tags (e.g., "provide NNS(clients) with"), uniformly distributes the pattern's frequency among all supersenses of the NN or NNS words. Then by re-grouping and re-ranking the semantically-motivated patterns, *GRASP* finds not only the grammatical contexts but the most probable semantic roles of NN and NNS tags in these contexts. Sample of semantically subcategorized patterns is shown in Table II where semantic roles are in squared parentheses.

TABLE II
PATTERNS BEFORE AND AFTER SEMANTIC ROLE LABELING

Query	Before	After
provide with	provide NNS with (394)	provide NNS[PERSON] with (252)
		provide NNS[GROUP] with (43)
provide to	provide NN to (325)	provide NN[COMMUNICATION] to (65)
		provide NN[ACT] to (63)

IV. EXPERIMENTS

A. Experimental Settings

We used British National Corpus (BNC) as our underlying large-sized general corpus *C*. It is a 100 million word collection of samples of written and spoken British English from a wide range of sources. We exploited GENIA tagger developed by Tsujii Laboratory to obtain the lemmas, PoS tags and shallow parsing results of *C*'s sentences. After lemmatizing and syntactic analyses, all sentences in BNC (approximately 5.6 million sentences) were used to build up inverted files and used as examples for extracting grammar patterns. As for bilingual parallel data, we used Hong Kong Parallel Text (LDC2004T08) assuming the first language of the language learners is Chinese. We leveraged CKIP Chinese segmentation system (Ma and Chen, 2003) to word segment the Chinese sentences within.

B. Interesting Patterns GRASP Extracted

In this subsection, we examine some grammar-like patterns generated by *GRASP*. Take monolingual query "make up" for

example. *GRASP* identified its four lexico-grammatical patterns with different associated senses: "make up PRP$[3] NN[COGNITION]" (e.g., "make up his mind"), "make up IN(for) DT" (e.g., "make up for the") for the sense *to compensate*, "NNS WDT make up" (e.g., "groups that make up") and passive "make up IN(of) NNS[PERSON]" (e.g., "made up of representatives") for the sense *to constitute*, and "make up DT NN[COMMUNICATION]" (e.g., "make up the story") for the sense *to fabricate*. It is challenging for collocation finders to obtain such patterns or usages since they usually do not accommodate multi-word queries, let alone finding the prepositions following a verbal phrase like "make up". Due to *GRASP*'s flexibility in the word order of the query in extracted patterns, it tolerates mis-ordered query words. Take the Chinese-ordered query "1990 Jan. 20" for example. The grammar pattern "IN Jan. 20 , 1990 , DT" (e.g., "On Jan. 20, 1990, the") *GRASP* yielded provides not only the common way to put dates in English sentences but the right order.

As for the cross-lingual mode, *GRASP* accepted Chinese queries like "打擊犯罪" (fight crime) and returned the characteristic syntax-based patterns anchored with their confident English translations: "fight crime", "combat crime" and "crack down on crime". EFL learners would benefit from cross-lingual *GRASP* in that it helps them to learn correct and yet versatile translations of the first-language queries, bypassing the erroneous user-nominated English queries because of first-language interference, as well as those translations' grammatical contexts. Take the Chinese query "學習知識" (*acquire knowledge*) for instance. *GRASP* responded with its diverse translation equivalents "acquire knowledge", "acquire the knowledge of", "learn skills" and so on, *excluding* the miscollocation "learn knowledge" commonly seen in English writing from Chinese learners.

C. Evaluation Results

To carefully control the variables in assessing the effectiveness of the thesaurus index structure *GRASP* provides for usage learning and navigation, we introduced monolingual *GRASP*[4] *alone* to EFL learners and they were taught on how to use *GRASP* for their benefits. Two classes of 32 and 86 first-year college students learning English as second language participated in our experiments. They were asked to perform a common language learning practice: sentence translation/composition, comprising two tests of pretest and posttest. In our experiments, pretest was a test where participants were asked to complete English sentences with their corresponding Chinese sentences as hints, while posttest was a test where, after utilizing traditional tools like dictionaries and online translation systems or *GRASP* in-between pretest and posttest to learn the usages of collocations/phrases in a candidate list provided by us, participants were also asked to complete the English translations of the Chinese sentences. In both the tests, there were exactly the same 15 to-be-finished test items, English translations with Chinese sentences, only with different orders. Each test item contains one frequent collocation/phrase based on the statistics from BNC corpus.

As mentioned above, a candidate list of 20 frequent collocations and phrases in BNC was provided for learning between tests. Participants were asked to concentrate on learning the contexts of the senses of the English collocations/phrases (e.g., "place order") specified by their Chinese counterparts (e.g., "下訂單"). To evaluate *GRASP*, half of the participants used *GRASP* for learning and the other half used traditional learning approach such as online dictionaries or online translation system (i.e., *Google Translate* and *Yahoo! Babel Fish*).

We summarize the averaged scores of our participants on pre- and post-test in Table 3 and 4 where *GRASP* stands for the (experimental) group using *GRASP* and *Trad* for the (controlled) group using traditional tools, and "ALL" denotes all students in the group, "UH" the upper half of the group in scores and "BH" the bottom half. As suggested by Table III and IV, the partition of the classes was quite random in that the difference between *GRASP* and *Trad* was insignificant under pretest and the index structure imposed by *GRASP* on words' contexts was helpful in language learning. Specifically, in table III *GRASP* helped to improve students' achievements on completing/composing the English sentences by 15.5% (41.9-26.4). Although students also performed better after consulting online dictionaries or translation systems by 5.6% (32.7-27.1), *GRASP* seemed to help students with more margin, almost tripled (15.5 vs. 5.6). Encouragingly, if we look closer, we find that both UH and BH students benefited from *GRASP*, from score 34.4 to 48.0 (+13.6) and from score 18.3 to 35.7 (+17.4), respectively. This suggests that the effectiveness of *GRASP* in language learning do not confine to certain level of students but crosses from high-achieving students to low-achieving.

TABLE III
THE PERFORMANCE ON PRETEST AND POSTTEST OF THE 1ST CLASS

	pretest (%)			**posttest (%)**		
	All	UH	BH	All	UH	BH
GRASP	26.4	34.4	18.3	**41.9**	**48.0**	**35.7**
Trad	27.1	34.2	19.9	32.7	33.4	32.0

TABLE IV
THE PERFORMANCE ON PRETEST AND POSTTEST OF THE 2ND CLASS

	pretest (%)	**posttest (%)**
GRASP	43.6	**58.4**
Trad	43.8	53.4

The helpfulness of *GRASP* was observed in another class (see Table IV). Class-to-class, in spite of the fact that the pretest performance of the 2nd class was much better than that

[3] PRP$ stands for a pronoun or a possessive.
[4] The system we introduced is at http://koromiko.cs.nthu.edu.tw/*GRASP*/

of the 1st class, the *GRASP* group of this high-achieving class *still* outperformed the *Trad* group (58.4 vs. 53.4), another indicator that the assistance of *GRASP* system is across different levels of students in language learning. Even in this comparatively high-performing class, the *GRASP*'s gain (58.4-43.6=14.8) is one third of the original pretest score (i.e., 43.6) and the gain is more than 1.5 times larger than *Trad*'s gain (53.4-43.8=9.6), suggesting that *GRASP* is much more effective and efficient in language learning than traditional lookup methods, mostly attributed to *GRASP* general-to-specific categorized usages, contexts, or phraseologies of words.

V. Conclusions and Future Work

Many avenues exist for future research and improvement of our system. For example, an interesting direction to explore is the effectiveness of our fully capable *GRASP*, responding to both monolingual and cross-lingual queries, in language learning. Additionally, we would like to examine the possibility of constructing a grammar checker based on our *GRASP* lexical-grammatical patterns. Yet another direction of research is to apply the *GRASP* framework to different languages and to associate the *GRASP*-extracted patterns in different languages for syntax-based machine translation system.

In summary, we have introduced a framework for learning to impose general-to-specific thesaurus index structures, comprising recurrent grammar patterns and their predominant lexical realizations, on queries' contexts. The characterizing context index structures assist users such as lexicographers and language learners in two ways: the generalization in patterns accelerates the navigation through different usages and the instantiations of patterns, i.e., lexical phrases, provide phraseological tendencies. We have implemented and evaluated the framework as applied to CALL, especially in second language writing. Extracted syntactic patterns have been shown to go beyond the collocations from common collocation finders and resemble phrases in grammar books. And we have verified (in two separate evaluations) that our hierarchical index structures on words' contextual regularity help the process of language learning.

References

[1] M. Benson, "Collocations and idioms," in Robert Ilson (Ed.), *Dictionaries, Lexicography and Language Learning*, 1985.

[2] M. Benson, E. Benson and R. Ilson, *The BBI Combinatory Dictionary of English. A Guide to Word Combinations*, 1986.

[3] W. Cheng, C. Greaves, and M. Warren, "From n-gram to skipgram to concgram," *Corpus Linguistics*, 11 (4), 2006.

[4] Y.C. Chang, J.S. Chang, H.J. Chen, and H.C. Liou, "An automatic collocation writing assistant for Taiwanese EFL learners: a case of corpus-based NLP technology," *Computer Assisted Language Learning*, 21 (3), 2008.

[5] P. Durrant, "Investigating the viability of a collocation list for students of English for academic purposes," *English for Specific Purposes*, 28 (3), 2009.

[6] S. Feldman, M. Marin, J. Medero, and M. Ostendorf, "Classifying factored genres with part-of-speech histograms," in *Proceedings of NAACL,* 2009.

[7] C.J. Fillmore, P. Kay, and M.K. O'Connor, "Regularity and idiomaticity in grammatical constructions: the case of *let alone*," *Language* 64, 1988.

[8] J.R. Firth, "Modes of meaning," *Papers in linguistics*. Oxford: Oxford University Press, 1957.

[9] M. Gamon, C. Leacock, C. Brockett, W.B. Dolan. J.F. Gao, D. Belenko, and A. Klementiev, "Using statistical techniques and web search to correct ESL errors," *CALICO*, 26(3), 2009.

[10] J.Y. Jian, Y.C. Chang, and J.S. Chang, "TANGO: Bilingual collocational concordance" in *Proceedings of ACL,* 2004.

[11] J.H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," in *Proceedings of EMNLP,* 2007.

[12] A. Kilgarriff, P. Rychly, P. Smrz, and D. Tugwell, "The sketch engine," in *Proceedings of EURALEX,* 2004.

[13] K. Kita and H. Ogata, "Collations in language learning: corpus-based automatic compilation of collocations and bilingual collocation concordance," in *Computer Assisted Language Learning*, 10 (3), 1997.

[14] P. Koehn, F.J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of NAACL/HLT,* 2003.

[15] M. Lewis, "Language in the Lexical Approach," in M. Lewis (Ed.), *Teaching Collocation: Further Development in the Lexical Approach*, 2000.

[16] L.E. Liu, *A corpus-based lexical semantic investigation of verb-noun miscollocations in Taiwan learners' English*, PHD dissertation, 2002.

[17] I.S.P. Nation, *Learning Vocabulary in Another Language*. Cambridge: Cambridge Press, 2001.

[18] N. Nesselhauf, "The use of collocations by advanced learners of English and some implications for teaching," in *Applied Linguistics*, 24 (3), 2003.

[19] F. Smadja, "Retrieving collocations from text: Xtract," *Computational Linguistics*, 19(1), 1993.

[20] M. Stubbs, "Two quantitative methods of studying phraseology in English," *Corpus Linguistics* 7(2), 2002.

[21] M. Stubbs, 2004. http://web.archive.org/web/20070828004603/http://www.uni-trier.de/uni/fb2/anglistik/Projekte/stubbs/icame-2004.htm.

[22] D. Wible and N.L. Tsao, "StringNet as a computational resource for discovering and investigating linguistic constructions," in *Proceedings of NAACL,* 2010.

[23] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the Annual Meeting of the ACL,* 1995.

Clause Boundary Identification using Classifier and Clause Markers in Urdu Language

Daraksha Parveen, Ratna Sanyal, and Afreen Ansari

***Abstract*—This paper presents the identification of clause boundary for the Urdu language. We have used Conditional Random Field as the classification method and the clause markers. The clause markers play the role to detect the type of sub-ordinate clause, which is with or within the main clause. If there is any misclassification after testing with different sentences then more rules are identified to get high recall and precision. Obtained results show that this approach efficiently determines the type of sub-ordinate clause and its boundary.**

***Index terms*—Clause marker, conditional random field.**

I. Introduction

CLAUSE boundary identification is a useful technique for various Natural Language Processing (NLP) applications. This is a method of specifying the beginning and ending of main and subordinate clause. Clauses are structural unit which have verbs with its arguments, adjuncts etc. There are 8 types of subordinate clause: *Complementizer, Relative Participle, Relative, Temporal, Manner, Causality, Condition* and *Nominal*. First three types of clauses are more syntactic while remaining five clauses are more semantic in nature.

Numerous techniques are used to recognize clause boundaries for different languages where some are Rule based [Harris 1997; Vilson 1998] and others are Statistical approaches using machine learning techniques [Vijay and Sobha, 2008]. A rule based clause boundary system has been proposed as preprocessing tool [Harris 1997] for bilingual alignment parallel text. In another pioneering work, a rule based system has been used which reduces clauses to noun, adjective or an adverb [Vilson 1998]. Identification of clauses for English language has been performed in an earlier research [Sang and Dejean, 2001]. A hybrid approach for clause boundary identification uses Conditional Random Fields (CRF) and rules, error pattern analyzer used to correct the false boundaries [Vijay and Sobha, 2008]. The clause identification for Tamil language shows 92.06% and 87.89% for precision and recall respectively, which in turns give the F-measure as 89.04%. The clause boundary identification has also been done for Bengali Language [Ghosh et. al. 2010]. CRF based statistical techniques are used to identify the type of clauses. The clause identification system gives the precision as 73%.

A basic clause identification system has been developed [Ejerhed 1988] for improving American Telephone & Telegraph (AT&T) text to speech system. This was used in English/Portuguese machine translation system. Clause spitting is also needed for the text to speech, which can be done by using conditional random fields' technique [Nguyen et.al. 2007]. In Korean language, analysis of dependency relation among clauses is very critical part. Kernel method [Kim et. al. 2007] is used to detect the clause boundaries. In Japanese language, there is no distinct boundary information to detect clauses; ambiguity can be minimized using rule based system [Fujisaki et.al. 1990].

In our present work, a hybrid approach is proposed that uses both techniques i.e. rule based and machine learning to build an identifier for different clause boundaries of Urdu language. We have applied the Conditional Random Fields (CRF). We have categorized the different types of sub ordinate clauses on the basis of clause markers. The POS tagger and Chunker [Pradeep et. al. 2007] are used to prepare the parts of speech and chunked tagged data as the inputs, where linguistic rules are taken as features. To the best of our knowledge, no work on identification of clauses for Urdu language is reported.

Henceforth presented details are divided into the following sections. We have given the introduction with related work in section 1. The methodology with clause markers, Clause Boundary Annotation Convention, Preprocessing, classification with features and rules are discussed in section 2. In the example sets, the Urdu sentences are translated in English for the easiness of the readers who are not familiar in Urdu. The algorithms for different phases are given in section 3. Section 4 shows the result of clause identification for Urdu language using this algorithm. Section 5 comprises the conclusion and finally reference section is included at the end.

II. Methodology

We have prepared the corpus for Urdu language. POS tagging and chunking are the preprocessing steps which have been done manually here, so contain a great accuracy. The POS and chunked tagged corpus has been considered as input data. Initially machine learning approach is applied, within which linguistic rules are used. Through this, clause boundary

Manuscript received November 2, 2010. Manuscript accepted for publication January 12, 2011.

The authors are with Indian Institute of Information Technology - Allahabad, India (e-mail: daraksha.parveen3022@gmail.com; rsanyal@iiita.ac.in; afreen.aa@gmail.com).

is recognized from input Urdu corpus. Now, if there is any misclassification, correction is done through additional linguistic rules. The work flow of identification of clauses is shown in Fig. 1.

We have used the CRF techniques as modeling in the learning phase and inference in the classification. This is a sequential classification technique which is taking care of many correlated features like in Maximum-entropy and a variety of other linear classifiers including winnow, AdaBoost, and support-vector machines [Sha et.al. 2003]. CRF gives more beneficial results than HMMs on a part-of-speech tagging task [Lafferty et.al. 2003]. Hidden Markov Model (HMM) needs to enumerate all possible observation sequences. This is not practical to represent multiple interacting features or long-range dependencies of the observations. Also it has very strict independence assumptions on the observations [Kelly et.al. 2009].

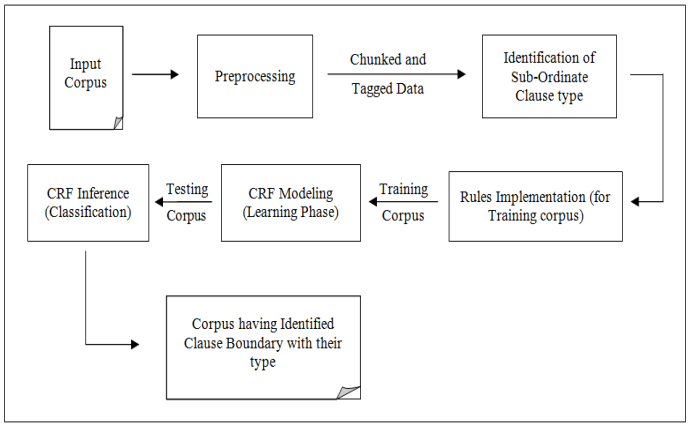


Fig. 1. Work flow for the identification clauses.

CRF uses the conditional probability *P (label sequence* **y** | *observation sequence* **x***)* rather than the joint probability *P*(**y, x**) as in case of HMM. It specifies the probability of possible label sequences **y** for a given observation sequence **x**. CRF allows arbitrary, non-independent features on **x** while HMM does not. Probability of transitions between labels may depend on past and future observations.

The shallow parsing uses special kind of CRF technique where all the nodes in the graph form a linear chain. In this type of graph, the set of cliques C (a graph in which every two subset of vertices are connected to each other) is just the set of all cliques of size 1 (i.e. the nodes) and the set of all cliques of size 2 (the edges). This technique has two phases for clause boundary identification:

1. *Learning*: Given a sample set X containing features $\{x_1, \ldots, x_N\}$ along with the set of values for hidden labels Y i.e. clause boundaries $\{y_1, \ldots, y_N\}$, learn the best possible potential functions.
2. *Inference*: For a given word there is some new observable x, find the most likely clause boundary y* for x, i.e. compute (exactly or approximately):

$$y^* = \arg \max_{y} P(y|x) \quad (1)$$

For this, an undirected and acyclic graph formed which contains the set of nodes $\{x_i\} \cup Y$ ($\forall x_i \in X$), adopts the properties by Markov, is called conditional random fields (CRFs). Clause Boundary Detection is a shallow parsing technique so, CRF is used for this.

A. Clause Markers

Clause markers are words or a group of words, like *now* and *well* in English, which helps in making the relation between the sentences. They are also used in combining two Urdu sentences as shown below.

(i) *[Ram ghar aya] aur [khana kha kar so gaya]*
[Ram came home] and [fell asleep after eating dinner]

As discussed earlier there are 9 types of subordinate clause. There are clause markers corresponding to these subordinate clauses which are more syntactic. For relative participle clauses, clause markers are *jo vo* (وہ جو), *jisne usne* (اُسنے جسنے), *jinhe* (جنھےں) etc. In the relative participle clause *vo* will always occur with *jo* as correlation [Butt et.al. 2007] as shown below

(ii) *[**Jo** ladka kal aya tha][vo ram hai]*
[That boy [who came yesterday] is Ram]

(iii) *[**vo** ladka [**jo** kal aya tha] ram hai]*
[Ram is the guy][who came yesterday]

Similarly, conditional clause markers, complementizer clause markers, and relative clause markers are shown below as boldface letter sequentially.

(iv)

*[**Agar** kal tum nahi aye] [**to** mein khane ke baad so jaunga]*
Conditional clause

[If u do not come tomorrow][then I'll sleep after eating dinner]

(v) *[Ram ne kal kaha tha] [**ki** vo ghar ja raha hai]*
Complementizer clause

[Ram said yesterday] [that he is going home]

(vi) *[Main [khana kha kar] so gaya]*
Temporal clause

[I fell asleep after dinner]

(vii) *[**jisse** mujhe kaam hai] [vo kahan hai]*
Relative clause

[Where is that person] [With whom I have a work]

B. Clause Boundary Annotation Convention

For main clause, clause boundary annotations are shown below where the symbols CL, B, M, and E are clause, beginning, main, and ending respectively.

```
CL = B_M
CL = E_M
```

For subordinate clause, clause boundary annotations are shown below where symbol 'Sub' indicates sub-ordinate clause.

```
CL = B_Sub_Type
CL = E_Sub_Type
```

Annotations for sub-ordinate clause types are shown below.

```
RELP: For Relative Participle
COMP: For Complementizer
COND: For Conditional
TMPR: For Temporal
CAUS: For Causal
RELC: For Relative
NOML: For Nominal
MANR: For Manner
```

C. Preprocessing

In the preprocessing stage, at first tagger is applied on the tokenized corpus to get tagged data and then chunker is applied to obtain chunked and tagged data (see Fig. 9). Further processing will be done on these tagged and chunked data. Sentence Boundaries are not given in the preprocessed data.

POS and Chunked tagged data are shown in Table I. There are three columns where first column comprises of tokens, second of tags for corresponding token and third contains chunking information. Here 'B' corresponds the beginning of phrase and 'I' to the words which are in a phrase.

TABLE I
POS AND CHUNKED TAGGED DATA

Tokens	Tags	Chunking
اَن‌ڈین	PRPN	B-NP
پیک	NN	I-NP
مشہور	ADJ	O
ہے	VAUX	B-VP
چہ اں	CC	B-CCP
پہ اڑوں	NN	B-NP
پر	PSP	O
لمبے	ADJ	O
وقت	NN	B-NP
تک	PSP	O
گھومنے	VB	B-VP
کے	PSP	O
ماہرین	ADJ	O
چڑھائی	ADJ	O
چڑھ	VB	B-VP
کر	VB	I-VP
دلی	ADJ	O
مراد	NN	B-NP
پوری	ADJ	O
کرتے	VB	B-VP
ہیں	VAUX	I-VP
۔	SYM	O

D. Classification

Sequence labeling classification technique is applied in the clause boundary identification. Clause Identification has been done by using linguistic rules which do not depends upon sentence boundaries. Classification technique requires features, training data set and testing data set. As discussed in sec. 2.1, classification has two phases, learning and inference. In learning phase, modeling takes place by taking training dataset as an input while in inference phase; classification of test data set takes place with the help of model obtained from learning phase.

1) Features

In this CRF technique linguistic rules are used as features for which different length of windows, comprises of words, are formed that depend on these linguistic rules. For example, in case of relative clause identification in Urdu language, clause beginning and ending are identified via rule1 and rule2 respectively.

RULE_1:

If the current word is any relative clause marker and next word is any of the POS tags verb, pronoun, adjective, noun then the next word is marked as beginning of clause boundary as shown below

```
Position 0: Relative clause marker
Position 1: Verb or Adjective or noun or pronoun
Then 0 should be marked as beginning of
subordinate clause of type relative.
```

Where position 0 indicates the current word and position 1 is the next word.

RULE_2:

If the current word is any verb auxiliary and next word is any symbol then current word is end of corresponding subordinate clause boundary as shown below

```
Position 0: Verb phrase or Verb auxiliary
Position 1: any symbol or phrase
Then 0 should be marked as end of above
subordinate clause.
```

2) Handling Misclassification

There is a chance of misclassification in the clause boundary ending. If there is any misclassification then correction is done through linguistic rule, which means priority is given higher to the linguistic rules.

III. Algorithm for Different Phases

A. Preparation of the Training Corpus

Step 1: First check whether a word W coming is a clause marker or not. If it is, then detect which type of clause it is.

Step 2: Implement those rules (defined as in sec.2.4.1) which is related to above type of clause which is detected in step 1. Then through these rules find the clause beginning and ending of that clause.

B. CRF Modeling (Learning Phase)

Step 1: Parse the prepared training corpus and assign $f_1, f_2, f_3, \ldots, f_m$ to those words which follows rule 1, rule 2, and rule 3… respectively.

Step 2: Make a matrix T of size M×N where,

M = no. of features ($f_1, f_2, f_3, \ldots, f_m$)

N = no. of classes (Clause beginning, Clause ending, not boundary)

Matrix is made by parsing the corpus in which,

T_{ij}=1, if a word follow rule i and belong to class j

T_{ij}=0, if not so

In this matrix we go on incrementing every time in T_{ij}, if another word follows the same.

C. CRF Testing

Step 1: Make a matrix J of size M×1 for each word where,

M = no. of features.

J_{i1} = 1, if a word follows rule i

J_{i1} = 0, if it does not follow

Step 2: Find matrix C of size 1×N

$$C_{1\times N} = J^{T}_{M\times 1} \times M_{M\times N} \quad (2)$$

Step 3: Assign that class to a word which has a maximum value in matrix C.

IV. Results and Discussion

The system is tested upon a corpus which consists of Urdu language dataset. The dataset comprises of different types of subordinate clause which is POS tagged and chunked. Results are shown in Table II which contains the information of clause boundary beginning and ending where B-SUB indicates the beginning of sub-ordinate clause while E-SUB is for ending of sub-ordinate clause. We have obtained the result using clause markers through which we can easily detect the type of subordinate clause. Evaluation of our system's performance is done by calculating the precision and recall as shown in Table III.

TABLE II
Output showing clause boundary beginning and ending

Tokens	Tags	Chunking
اس	PRN	B-NP
وقت	NN	I-NP
ادہ‌ی‌ز	ADV	O
تر	ADV	O
باشندگان	ADJ	O
تر	ADV	O
غلام	NN	B-NP
تھے	VAUX	B-VP
جو	CC	B-CCP <Cl=B-SUB-RELP>
باغات	NN	B-NP
ں‌ی‌م	PSP	O
کام	NN	B-NP
کرتے	VB	B-VP
تھے	VAUX	I-VP <Cl=E-SUB-RELP>
۔	SYM	O

Table III shows the comparison between different ratios of corpus taken for training and testing purpose. In the corpus (developed for this work only), there are 139 different sentences with POS and Chunked tagged related to tourism domain. It is a 3-fold cross Validation represented by set-1, set-2 and set-3.

TABLE III
Comparison of Different Ratios of Training and Testing Corpus

Training-Testing	Precision (%)	Recall (%)
90%-10%		
Set – 1	89.2	90.0
Set – 2	88.6	89.5
Set – 3	87.5	88.9
Average value	88.4	89.5
Standard Deviation	0.705	0.451
80%-20%		
Set – 1	85.2	86.7
Set – 2	86.0	87.1
Set – 3	85.5	87.5
Average value	85.6	87.1
Standard Deviation	0.331	0.327
70%-30%		
Set – 1	82.3	84.0
Set – 2	82.6	83.9
Set – 3	82.0	84.1
Average value	82.3	84.0
Standard Deviation	0.245	0.082

Our system works very efficiently on the similar sentences shown below

Relative sub-ordinate clause

(i) [پیرس نے اس علاقہ کی طرف کوئی توجہ نہ‌یں دی] [جسسے حالت بد سے بدتر ہ‌وتی گئی]-

[Paris did not pay any attention to that area] [due to which the condition get worsened.]

Relative Participle sub-ordinate clause

(ii) [اس وقت زیادہ تر باشندگان غلام تھے] [جو باغات میں کام کرتے تھے]

[That time mostly peoples were slaves] [who worked in gardens.]

Complementizer clause

(iii) [کورٹ کے حق میں ایک بات اور ہ‌و گئی] [کہ یہ محض اتفاق تھ ||

[Court was in favor of something] [that was just a coincidence.]

The problem for detecting the clause ending is coming for the following types of sentences.

Relative Participle sub-ordinate clause

(i) [جب برٹش ٹاسک فورس جو چھگڑا کے اوائل میں ہی روانہ کر دی گئی تھی] ، [تب متعدد مواقع پر بات چیت اور بیچ بچاؤ ناکامیاب رہے]

[When the British Task Force early in the dispute had been dispatched], [then they discuss various opportunities and failed to get rescue]

Temporal sub-ordinate clause

(ii)[میں [اکھان کھا کر] سو گیا]

[I fell asleep after dinner]

Manner Sub-ordinate clause

(iii)[میں ورجش کروں گا [جیسا کہ مجھے سکھایا گیا] ہے]

[I'll do the exercises [as I've been taught]]

After analyzing the above sentences, we have found that the sentences where the distance of clause beginning and ending is significantly large, our system is unable to detect the clause ending correctly as shown above in first sentence. Here, big braces show the actual clause beginning and ending whereas our system is unable to detect the clause ending. For those sentences which are semantic in nature, it is difficult for our system to detect clause ending and beginning as shown above in second, third and fourth sentences.

V. Conclusion

In this paper Conditional Random Fields are used for classification of clause boundary beginning and ending and also detecting the type of subordinate clause. Here, linguistic rules are given higher priority, hence misclassification is corrected via these rules. Limitation with CRFs is that it is highly dependent on linguistic rules. Missing of these rules may lead to wrongly classified data. An improvement can be achieved in the proposed clause boundary identifier by including more sophisticated linguistic rules, clause markers for different subordinate clauses and also for those clauses which are embedded in the main clause. For future work, clause boundaries detection can be done on those sentences where distance between clause beginning and ending is significantly large and also where the sub-ordinate clauses in the sentences are semantic in nature. More linguistic rules are being identified and will be apply. This work is in progress.

Acknowledgement

The authors gratefully acknowledge the support from Indian Institute of Information Technology Allahabad and Technology Development in Indian Languages, funded by MCIT (Govt. of India).

References

[1] M. Butt, T.H. King, and S. Roth, "Urdu correlatives: theoretical and implementational issues," in *Proceedings of the LFG07 Conference*, CSLI publication, 2007, pp. 107-127.

[2] E. Ejerhed, "Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods," in *Proceedings of the 2nd Conference on Applied Natural Language Processing*, Austin Texas, 1988, pp. 219-227.

[3] H. Fujisaki, K. Hirose, H. Kawai, and Y. Asano, "A System for synthesizing Japanese speech from orthographic text," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing ICASSP-90*, vol.1, 1990, pp. 617-620.

[4] A. Ghosh, A. Das, and S. Bandyopadhyay, "Clause Identification and Classification in Bengali," *in Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP, 23rd International Conference on Computational Linguistics (COLING)*, Beijing, August 2010, pp. 17-25.

[5] V. P. Harris, "Clause Recognition in the Framework of Alignment," Mitkov, R., Nicolov, N. (eds.) *Recent Advances in Natural Language Processing*, John Benjamins Publishing Company, Amsterdam/Philadelphia, 1997, pp. 417-425.

[6] D. Kelly, J. McDonald, and C. Markham, "Evaluation of threshold model HMMS and Conditional Random Fields for recognition of spatiotemporal gestures in sign language," in *Proceedings of the 12th international conference Computer Vision Workshops (ICCV Workshops 2009),* 2009, pp. 490-497.

[7] S. Kim, S. Park, S. Lee, and K. Kim, "A Feature Space Expression to Analyze Dependency of Korean Clauses with a Composite Kernel," in *Proceedings of the 6th International Conference Advanced Language Processing and Web Information Technology (ALPIT 2007)*, 2007, pp. 57-62.

[8] J.D. Lafferty, A. McCallum, and F.C.N. Pereira, "Conditional Random Fields: Probabilistic Models For Segmenting and Labeling Sequence Data," in *ICML '03 Proceedings of the Eighteenth International Conference on Machine Learning*, 2003, pp. 282-289.

[9] V. Nguyen, "Using Conditional Random Fields for Clause Splitting," in *Proceedings of the Pacific Association for Computational Linguistics*, University of Melbourne Australia, 2007.

[10] V.D. Pradeep, M. Rakesh, and R. Sanyal, "HMM-based Language independent POS tagger," in *Third Indian International conference on Artificial Intelligence IICAI 2007,* 2007.

[11] E.F.T.K Sang and D. Herve, "Introduction to CoNLL-2001 shared task: clause identification," in Walter Daelemans and Remi Zajac (eds.) *Proceedings of Conference on Computational Natural Language (CoNLL 2001),* Toulouse, France, 2001, pp. 53-57.

[12] F. Sha and F. Pereira, "Shallow Parsing with Conditional Random Fields," in *NAACL '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Volume 1, pp. 134-141, 2003.

[13] R.S.R. Vijay and L.D. Sobha, "Clause Boundary Identification Using Conditional Random Fields," in *Lecture Notes in Computer Science, Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, Springer-Verlag, 2008, pp. 140-150.

[14] J.L. Vilson, "Clause Processing in Complex Sentences," in *Proceedings of the First International Conference on Language Resource and Evaluation*, vol. 1, 1998, pp. 937-943.

External Sandhi and its Relevance to Syntactic Treebanking

Sudheer Kolachina, Dipti Misra Sharma, Phani Gadde, Meher Vijay,
Rajeev Sangal, and Akshar Bharati

***Abstract*—*External sandhi* is a linguistic phenomenon which refers to a set of sound changes that occur at word boundaries. These changes are similar to phonological processes such as assimilation and fusion when they apply at the level of prosody, such as in connected speech. External sandhi formation can be orthographically reflected in some languages. External sandhi formation in such languages, causes the occurrence of forms which are morphologically unanalyzable, thus posing a problem for all kind of NLP applications. In this paper, we discuss the implications that this phenomenon has for the syntactic annotation of sentences in Telugu, an Indian language with agglutinative morphology. We describe in detail, how external sandhi formation in Telugu, if not handled prior to dependency annotation, leads either to loss or misrepresentation of syntactic information in the treebank. This phenomenon, we argue, necessitates the introduction of a sandhi splitting stage in the generic annotation pipeline currently being followed for the treebanking of Indian languages. We identify one type of external sandhi widely occurring in the previous version of the Telugu treebank (version 0.2) and manually split all its instances leading to the development of a new version 0.5. We also conduct an experiment with a statistical parser to empirically verify the usefulness of the changes made to the treebank. Comparing the parsing accuracies obtained on versions 0.2 and 0.5 of the treebank, we observe that splitting even just one type of external sandhi leads to an increase in the overall parsing accuracies.**

***Index Terms*—Syntactic treebanks, sandhi.**

I. Introduction

In recent years, there has been a steady increase in awareness about the multi-fold importance of treebank corpora. This is evident from the number of syntactic treebanking projects for different languages that have been initiated and are currently ongoing. Although initial efforts such as the Penn treebank (PTB) [1] worked with constituency-based representations, many treebanking efforts in the last decade have preferred dependency-based representations. Two main reasons for this preference can be gathered from the literature. The first reason is that for languages that are relatively free word-order, dependency-based representation has been observed to work better [2], [3], [4], [5]. The second is that the simplicity of dependency-based representation makes it amenable for a variety of natural language processing applications [6], [7].

A dependency annotation scheme inspired by Paṇinian theory was proposed for syntactic treebanking of Indian languages (ILs) which are both morphologically rich and relatively free word-order [8]. This scheme has hitherto been applied to three Indian languages: Hindi, Bangla and Telugu. The first versions of all three treebanks were released for the shared task on IL parsing held as part of ICON-2009[1]. While the Hindi treebanking effort has matured considerably and the treebank is being developed at a stable pace [9], [10], [11], the Telugu and Bangla treebanks are still at a very initial stage of development. In this paper, we discuss some issues in Telugu treebanking with special reference to a linguistic phenomenon known as *external sandhi*. We discuss how external sandhi formation in Telugu poses problems during syntactic annotation of Telugu sentences. We discuss how this language-specific issue necessitates the introduction of a sandhi splitting or segmentation stage in the generic annotation pipeline. As a preliminary step towards a sandhi split treebank of Telugu, we manually split all instances of one type of external sandhi widely occurring in the previous version of the treebank. We conduct an experiment to empirically verify the efficacy of sandhi splitting in the Telugu treebank for the task of NL parsing.

II. Related Work and Background

Begum et al. [8] proposed a dependency-based annotation scheme for syntactic treebanking of Indian languages. Currently, treebanks of three Indian languages[2], Hindi, Bangla and Telugu are being developed using this annotation scheme. All these three languages are morphologically rich and have a relatively free word order. The applicability of this scheme for syntactic annotation of a fixed word order language like English has also been studied to some extent [12], [13]. The annotation scheme is based on a grammatical formalism known as Computational Paṇinian Grammar (CPG), a brief introduction to which is given in the next section. Vempaty et al. [14] give a detailed account of the issues encountered in the application of this annotation scheme to the treebanking of Telugu along with the decisions taken to address each of

Manuscript received October 27, 2010. Manuscript accepted for publication January 14, 2011.

The authors are with the Language Technologies Research Centre, IIIT-Hyderabad, India (e-mail: {sudheer.kpg08, phani.gadde, mehervijay.yeleti}@research.iiit.ac.in, {dipti, sangal}@mail.iiit.ac.in)

[1]NLP Tools Contest on IL parsing. *http://ltrc.iiit.ac.in/nlptools2009/*

[2]HyDT-Hindi, HyDT-Bangla and HyDT-Telugu

them in the development of version 0.1 of the treebank. They also discuss a few syntactic constructions in Telugu which are of interest from the parsing perspective.

Telugu is a Dravidian language with agglutinative morphology [15]. Although all Indian languages in general are said to be morphologically rich and therefore have relatively free word order, there exist considerable differences among them with respect to their finer morphological properties. For instance, the morphology of Hindi, an Indo-Aryan language is said to be inflectional as one morph can be mapped to several morphemes. However, inflectional morphs in Hindi such as case-markers and auxiliaries are not bound to their stems. This is typically considered a property of languages with analytical morphology. Telugu, on the other hand, is characterized as having an agglutinative morphology. It must be noted that agglutination in its original formulation [16], refers to the property of a one-to-one mapping between morphs and morphemes. In Telugu, inflectional morphs (which include different kinds of auxiliary verbs and case-markers) are always bound to the stem resulting in highly synthetic word forms. The number of possible verb forms for a verb stem in Telugu therefore, is very high, aggravating the task of the morph analyzer. In addition, as we will show through example sentences in section IV, even full morphological words can fuse together in Telugu resulting in complex forms which are morphologically unanalyzable[3]. Such complexities in the morphology of Indian languages point towards the need for a more exact approach while typifying them similar perhaps, to the one espoused in Greenberg [17].

The notion of external sandhi in traditional Sanskrit grammars captures well the phenomenon of word fusion mentioned above. In section IV, we show how this phenomenon, if not addressed through sandhi-splitting, poses problems for syntactic analysis of Telugu sentences during treebanking. This phenomenon was not handled in the preliminary version of the Telugu treebank[4] released for the shared task on IL parsing at ICON 2009. Although the number of sentences in the Telugu treebank (1400 sentences for training, 150 sentences each for development and testing) was comparable to that of the Hindi and the Bangla treebanks[5], the average parsing accuracy for Telugu on both coarse and fine-grained datasets was much lower as compared to the other languages [18]. In fact, all the participating systems reported their lowest accuracies on the Telugu datasets. An analysis of the Telugu treebank was carried out in order to discover possible reasons for such low accuracies on the parsing task. As a result, two possible reasons were identified. One reason for the relatively low accuracies on the Telugu datasets was that there was a considerable difference of domain between the training set on the one hand, and the development and test sets on the other. Such ill-effects of domain differences on the parsing accuracies can be easily avoided by partitioning the treebank differently and are hence, not a source of worry during treebank development.

TABLE I
IL TREEBANK STATISTICS: A COMPARISON

Language	sentences	words / sentence	chunks / sentence
Hindi	1800	19.01	9.18
Bangla	1280	10.52	6.5
Telugu	1700	5.43	3.78

The second reason was the lower number of both words and chunks in Telugu sentences as compared to Hindi and Bangla (shown in Table I[6]). The shared task dealt with chunk parsing which means that dependencies are shown only among the chunks[19] in a sentence. In the case of Telugu, as the above table shows, different syntactic relations are possible with the same dependency structure leading to sparsity. Statistical parsers have to learn the same number of syntactic relations from relatively lesser structure in Telugu as compared to Hindi and Bangla. We show in section IV that the smaller number of chunks per sentence in Telugu is directly attributable to the phenomenon of external sandhi formation.

It must be mentioned that in terms of parsing accuracies, a similar pattern was observed in the case of Turkish at both the CONLL shared tasks on dependency parsing [20], [21]. Interestingly, Turkish is also an agglutinating language with relatively free word order. In fact, the agglutinative morphology of Turkish dictated the choice of the treebank architecture used in the development of the Turkish treebank [5]. In the Turkish treebank, the complex agglutinative word forms are represented as sequences of *inflectional groups* separated at derivational boundaries. Syntactic relations are represented between these inflectional groups rather than between word forms. This information about word structure is preserved because morphological features of intermediate derivations are cues for syntactic relationships. Thus, annotation of syntactic dependencies is done on top of a tier of morphological segmentation. It would be interesting to do a detailed comparison of the annotation schemes of the Turkish and Indian language treebanking efforts vis-a-vis the properties of these languages.

In another related work, external sandhi has been recently discussed in the context of building an automatic Sanskrit segmentizer [22]. In this work, two different approaches to automatic segmentation are explored, both of which perform with reasonable accuracy.

III. CPG FORMALISM

The CPG formalism is a dependency grammar inspired, as mentioned previously, by Paṇinian grammatical theory. In this

[3]It would not be correct to call instances of such fusion as 'word forms' as they fall outside the domain of morphology. We argue in section IV that they are a result of orthographic expression of prosodic processes.

[4]version 0.2

[5]Hindi: 1500 training, 150 development, 150 testing
Bangla: 980 training, 150 development, 150 testing

[6]Note that the comparison shown here is based on the datasets released for the shared task on parsing held at ICON 2009.

formalism, as in other dependency grammars, the syntactic structure of a sentence in a natural language consists of a set of binary asymmetric relations called *dependencies* between the 'words' (lexical items) in that sentence. A dependency relation is always defined between a *head* word and a *modifier* word that modifies the head. In the CPG formalism, the verb is treated as the head of a clause. Nouns denoting the participants in the activity denoted by the verb stem are treated as modifiers of the verb. The relation between a verb and its modifier is known as a *karaka* relation, a notion central to syntactic analysis in Paṇinian grammar. *Karaka* relations are syntactico-semantic relations that obtain between a verb and its modifier. Each participant of the activity denoted by a verbal stem is assigned a distinct *karaka*. For example, $k1$ or *karta* is a relation that picks out the participant most central to the activity denoted by the verb. There are six different *karaka* relations defined in Paṇinian grammar. In addition to *karaka* relations that obtain between verbs and their participants, dependency relations can also exist between pairs of nouns (genitives), between nouns and their modifiers (adjectival modification, relativization), between verbs and their modifiers (adverbial modification including clausal subordination). A detailed dependency label tagset encompassing all these different kinds of relations is defined in the annotation scheme based on the CPG formalism [23].

One important point of departure in CPG from other dependency grammars is that dependency relations may also be defined between groups of words known as *chunks*. A chunk is defined as a minimal, non-recursive structure consisting of a group of related words. Each chunk has a unique head word whose category determines the chunk type. This head word is modified by the other words in the chunk. In a chunk-based dependency representation, dependency relations are defined between chunk heads. Another important concept in CPG which relates to the notion of a chunk is the *vibhakti*. For a noun chunk, *vibhakti* is the post-position/suffixes occurring after the noun which encodes information about case-marking and thematic roles (via the notion of *karaka* which is syntactico-semantic). Similarly, in the case of a verb chunk, the verbal head may be followed by auxiliary verbs which may remain as separate words or combine with the head as suffixes depending on the morphology of the language. This information following the head is collectively called the *vibhakti* of the verb. The *vibhakti* of a verb chunk encodes information about the tense, aspect and modality (TAM) as well as agreement features of the verb. Both these kinds of *vibhakti* have been shown to be crucial in NLP applications such as parsing [24]. In fact, even during annotation, nominal *vibhaktis* serve as cues to identify the *karaka* relation that can be assigned to the noun.

IV. External Sandhi and its Relevance to Syntactic Annotation

The origins of the notion of *sandhi* can be traced back to the seminal work of theorists such as Paṇini in the Indian linguistic tradition. Briefly stated, sandhi ('putting together') refers to a set of morpho-phonological processes that occur at either morpheme or word boundaries. These processes are captured as sandhi rules in traditional Sanskrit grammars which are based chiefly on the **avoidance of hiatus** and on **assimilation** [25]. It must be noted that sandhi is also reflected in the orthography of Sanskrit as the coalescence of final and initial letters. Two types of sandhi are identified in language, **internal sandhi** and **external sandhi**. Internal sandhi refers to word-internal morphonological changes that take place at morpheme boundaries during the process of word-formation. The internal sandhi rules in Sanskrit grammar apply to the final letters of verbal roots and nominal stems when followed by certain suffixes or terminations [25]. An example of internal sandhi in English would be the positional variation of the negative morpheme 'in-' to give the allomorph 'im-' when it is prefixed to words that begin with bilabial sounds (as in 'impossible'). Such processes lie obviously, within the domain of morphology. External sandhi, on the other hand, refers to processes that apply word-externally (across *word* boundaries). External sandhi rules in Sanskrit grammar determine the changes of final and initial letters of words [25]. Examples of external sandhi formation in English are the well-known cases of *wanna*/*hafta*/*gotta* contractions where the verb combines with the infinitival 'to' following it to give the contracted form. Note that external sandhi as seen in these examples need not always be reflected orthographically in English ('want to' while writing, but spoken as 'wanna').

The notion of sandhi formation is also well-known in modern linguistics. However, in much of the literature on this subject, sandhi is treated purely as a phonological process. That phonological processes occur across morpheme boundaries and not across word boundaries is the default situation in much of phonology. Cases where phonological rules apply across word boundaries, in other words, cases of external sandhi formation, have attracted special attention in generative phonology. In fact, this phenomenon is one of the central motivations for the theory of *Prosodic Phonology* [26], which formalizes the intervention of syntactic conditions (the relationships between words) in phonological matters. A similar notion discussed in the literature is that of *phonological phrase* which is defined as an organizational unit in phonology (parallel to syntactic phrase in syntax). A phonological phrase, therefore, is the domain within which external sandhi rules operate [27].

As discussed earlier in the case of English contractions, external sandhi can be limited to connected speech and need not be present in text. However, in Indian languages, especially Dravidian languages, sandhi (both internal and external) has a wide-spread occurrence and is also orthographically reflected most of the time. Sandhi phenomena in European languages have also been studied extensively [28] and external sandhi has been discussed as occurring prominently in Italian [29]. We refer to all such languages with high prevalence of external sandhi as Sandhi languages. External sandhi formation in

Sandhi languages leads to fusion of morphological words resulting in morphologically complex/unanalyzable forms. This poses a problem for all natural language processing applications such as POS-tagging, chunking, parsing, etc. that deal with written text. The task of tokenization becomes complex in these languages as tokens obtained through sentence splitting can contain more than one morphological word within them. Since external sandhi is a consequence of (orthographically visible) phonological processes occurring at the prosodic level, splitting such instances of sandhi can not fall within the purview of the morph analyzer. The task of splitting sandhi forms requires segmentation at a different level and should be treated as being distinct from morphological segmentation. Without this distinction between sandhi formation and other kinds of morphological changes, the task of morphological analysis in languages like Telugu becomes extremely complex. In the case of syntactic treebanking, if cases of external sandhi are not handled appropriately during tokenization, information about the syntactic relations that obtain between the words fused together due to external sandhi formation would be lost. This can be seen from the following Telugu examples.

```
(kanpu)_NP (warvAwa)_NP (wagina)_VGNF
childbirth after        appropriate
    (saMrakRaNa)_NP (lexanukoMdi)_VGF
    protection      Neg-Fin+think-Fin-Sandhi
'Think that there is no proper protection
    post-partum'
```

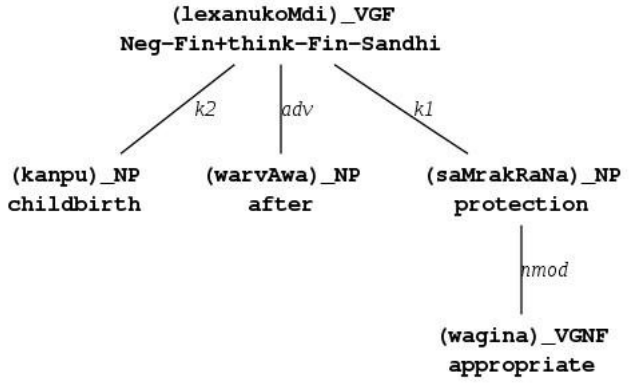


Fig. 1. Example dependency tree from the Telugu treebank without sandhi splitting.

In this example, the verb in the matrix clause is 'think' (Telugu stem: 'anukoVtam') which takes a clausal complement. The verb in the complement clause is a Negative finite verb ('lexu'). It can be noticed that in the above example, both the verbs are fused together as a result of external sandhi formation. If this issue is not handled prior to dependency annotation, dependencies would have to be incorrectly shown between the chunks in the sentence and the fused form ('lexanukoMdi') which is treated as the head of the sentence. This is linguistically inappropriate for the obvious reason that the information pertaining to the argument structure of both the verbs in this sentence is lost in such a representation. The dependency tree corresponding to this sentence from version 0.2 in Fig. 1 clearly shows this fallacy. The correct dependency tree for this sentence is also shown in Fig. 2 for comparison.

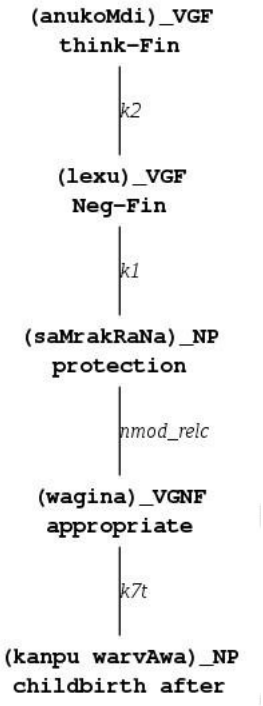


Fig. 2. Example dependency tree from the Telugu treebank with sandhi splitting.

The following example sentences too contain instances of external sandhi formation.

```
(rogaM)_NP (muxirina)_VGNF (pillani)_NP
disease     worsen         girl-Acc
    (xavAKAnAlo)_NP (cuparemi)_VGF
    hospital-Loc    show-Fin-emi-Sandhi
'Why dont (you) show this disease afflicted
    girl in a hospital?'

(ammakemi)_NP (wocakuMdeV)_VGF
Mother-Dat-emi-Sandhi strike-Fin
'Mother could not think of anything'
(literal: Anything did not strike mother)
```

The word 'emi' undergoes external sandhi in both these sentences. While in the first sentence, it gets fused to the predicate in that clause ('cuparu'), in the second sentence, it fuses with another argument ('ammaku') of the predicate. Corresponding to this difference in sandhi formation, there exists an interesting difference in the function of 'emi' which can be noticed from the translations provided. In the first sentence where it fuses with the verb, its function is similar to that of a question word (or clitic). In the second sentence, however, it functionally resembles a Negative Polarity Item (NPI). This correlation between sandhi formation and the

function of 'emi' can be seen as a direct evidence for the interaction between syntax and prosody. While syntactically annotating these sentences, if the sandhi form is not split, information about the syntactic relation of 'emi' to its head would be lost. In fact, this word/clitic 'emi' belongs to a paradigm of demonstrative pronouns in Telugu all of which can potentially exhibit similar behaviour. If the sandhi in each of these sentences is not split, syntactically important information such as argument structure would be either lost or misrepresented in the treebank.

In the next section, we describe how such sandhi forms in the previous version of the treebank were manually split leading to a new version of the treebank.

V. Sandhi Splitting in Syntactic Treebanking

The examples in the previous section show how external sandhi formation in Telugu, can lead to either loss or misrepresentation of syntactic information in the treebank. The sandhi forms in such sentences need to be split so that syntactic relationships involving tokens undergoing sandhi are accurately represented. In this section, we discuss the rationale for introduction of a distinct sandhi splitting stage in the annotation pipeline. We also describe how sandhi forms in the Telugu treebank were split to produce a new sandhi split version. However, we first give a brief overview of the annotation pipeline being followed for IL treebanking.

A. Annotation Pipeline for IL Treebanking

The annotation process followed to develop a treebank of CPG-based dependency structures for Indian languages consists of multiple steps (see Fig. 3). Sentences are tokenized to begin with. The tokens obtained at the end of this step are analyzed by a morph analyzer in the next step. At the third stage, the tokens in the sentence are POS-tagged which is followed by chunking at the fourth stage where tokens are grouped into chunks. As mentioned earlier, it is possible in this annotation scheme to annotate dependency relations between chunks (in fact, *heads* of chunks). This distinction between inter-chunk and intra-chunk dependencies is based on the observation that intra-chunk dependencies can be generated with high accuracy given the chunks and their heads (except in very few cases such as compounds, collocations). Thus, annotation of inter-chunk dependencies alone in phase 1 of the annotation would result in a chunk-level dependency treebank. This strategy also minimizes the time requirements of syntactic treebank development which is usually seen as a labor-intensive and time-consuming task. At the next stage in the pipeline, the dependency relations are annotated between the chunks. This is followed by post-processing in the form of quality checks and validation. The processing at each of these stages can be automated followed by human post-editing. Information obtained at each stage of processing is used by subsequent stages. Currently, processing at the first four stages, namely tokenization, morph analysis, POS-tagging and chunking, is being reliably done using highly accurate tools. The task of dependency annotation can also be automated using *vibhaktis* as cues for *karaka* assignment. However, it must be noted that *vibhaktis* can also be ambiguous which is why the task of *karaka* assignment is not always straight-forward. Therefore, it was decided that reliable annotation of syntactic dependencies can be achieved only through manual annotation.

In order to get an idea about the degree of occurrence of external sandhi, and also about the different kinds of sandhi possible, a detailed manual study of 600 sentences from the previous version of the Telugu treebank was done. We observed that there was no straight-forward method to identify sandhi forms in Telugu. The assumption that sandhi forms can be identified based on the output of the morphological analyzer is not correct. This is because forms for which the paradigm-based morph analyzer does not generate any analysis include, apart from cases of external sandhi, inflections of unknown words. In addition, the morph analyzer sometimes analyses sandhi forms incorrectly treating them as words. These observations suggest that splitting of sandhi forms should precede morph analysis. In fact, Sandhi splitting must be done as part of the tokenization step as external sandhi causes the fusion of tokens. Once the tokens in a sentence are obtained, all the other steps in the annotation process can be carried out without any changes. Fig. 3 shows the annotation pipeline with a sandhi splitting stage introduced prior to morph analysis. This additional stage would be necessary for all Sandhi languages.

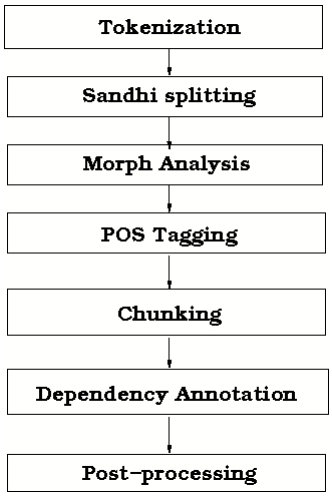


Fig. 3. Modified annotation pipeline for Sandhi languages such as Telugu.

B. Sandhi Splitting in the Telugu Treebank

In the course of the manual study, it was observed that *e*-demonstratives in Telugu (such as 'emi' in the examples from the previous section), undergo sandhi with neighbouring words most of the time. In this regard, they can be compared

TABLE II
DIFFERENT KINDS OF MODIFICATIONS MADE TO VERSION 0.2 OF THE TELUGU TREEBANK

Type of modification	# of modifications
POS-tag corrections	239
Chunk tag corrections	136
DepRel corrections	673
Sandhi splitting changes	179
All	1227

to what are known an *leaners* such as 'to' in English [27]. Another interesting observation about these elements is that their function seems to vary according to the part of speech of the word they lean on. Since this class of pronouns has a distinct form, it is possible to easily extract their instances from the treebank. All instances of external sandhi involving these demonstratives were extracted and split. Since this work is a preliminary exploration of external sandhi in the Telugu treebank, we restrict ourselves to manual sandhi splitting. However, in order to be able to split all types of external sandhi over the entire treebank, automating the task of sandhi segmentation is a mandatory requirement. It is expected that the sandhi rules in Telugu we encountered during the process of manual sandhi splitting would aid in the development of an automatic segmentizer.

In the process of manual sandhi splitting from the previous version of the treebank, we encountered POS-tag and chunk tag errors made by the automatic taggers which were corrected. Errors in annotation of syntactic dependencies (both attachment and relation label) from earlier phases of annotation were also corrected. The statistics about all these different kinds of modifications are given in Table II. The new version of the treebank resulting from these modifications is numbered as version 0.5[7].

VI. PARSING EXPERIMENT

Although treebanks can be used for a variety of purposes, the major impetus for treebanking in recent times, has come from the rapid developments in the area of data-driven natural language parsing. In fact, the relationship of sandhi formation with the syntax of Telugu discussed in this work was discovered in the light of a detailed analysis of the results of the NLP tools' contest for IL parsing at ICON 2009[8]. In order to empirically verify the usefulness of sandhi splitting in the treebank for syntactic parsing, we experiment by applying a data-driven dependency parser first, to sentences from the previous version of the treebank and then, to the new version in which sandhi splitting was manually done. A comparison of the parsing accuracies obtained using these two versions of the treebank would help us understand not only the effect of sandhi formation on Telugu parsing but also the efficiency of the design choices we made to address it in the new

[7]This new version of the treebank is released for the NLP tools' contest on IL parsing at ICON 2010. *http://ltrc.iiit.ac.in/nlptools2010/*

[8]*http://ltrc.iiit.ac.in/nlptools2009/*

TABLE III
DESCRIPTION OF THE DATASETS USED IN THE PARSING EXPERIMENTS

Dataset	Description
set-0	1600 sentences from treebank version 0.2
set-1	POS-tag, Chunk tag and DepRel error corrections
set-2	sandhi-splitting changes only
set-3	both changes

version of the treebank. In addition, as already mentioned in the previous section, modifications made to the previous version of the treebank include post-editing changes wherein the errors of the automatic POS-tagger and chunker are corrected and also, dependency corrections (both attachment and label corrections). For our parsing experiments, we created four different datasets each containing a different version of the same set of sentences. Set-0 contains sentences drawn from the previous version of the treebank. The sentences in set-0 are replaced by their post-edited (POS-tag, chunk tag and dependency relation corrected) versions to create set-1. Sentences in set-0 containing instances of external sandhi are replaced by their sandhi split versions to create set-2. Finally, set-3 is made up of sentences containing both post-editing and sandhi-splitting changes. The details of the datasets are briefly summarized in Table III.

Applying a data-driven parser to these different datasets, we tried to tease apart the influence of these different kinds of modifications on the parsing accuracy. We use the publicly available MaltParser [30] in this experiment with learner and feature model settings identical to those of the system that reported the highest accuracies for Telugu parsing at the NLP tools' contest 2009. In order to be able to pin-point the effect of the annotation changes on the parser performance and also, to normalize for sentence length and complexity, we ran the parser in cross-validation mode (10-fold) besides applying it to a test set of 150 sentences. Both these accuracies for each dataset are shown in table IV.

As shown in table III, set-0 is comprised of sentences drawn from the previous version of the treebank. Therefore, the accuracies obtained on set-0 are treated as the baseline accuracies in this experiment. It must be noted that the baseline accuracies obtained in our experiment using set-0 are considerably higher than the best accuracies reported on the same version of the treebank released for the NLP tools' contest shared task on parsing 2009 [18]. This difference in accuracies can be attributed solely to the way the treebank was partitioned to create the released datasets. The accuracies obtained on set-1 are slightly greater than the baseline accuracies. The increase in unlabeled attachment score (UAS) (both cross-validation and test set) is higher than the one in labeled attachment score (LAS). This difference between the accuracies obtained on set-1 and the baseline accuracies demonstrates the effect of correction of errors from the previous version of the treebank.

The accuracies obtained on set-2, although better than the baseline accuracies, are less than the accuracies obtained

TABLE IV
ACCURACIES ON THE FINE-GRAINED DATASETS

Dataset	Cross-Validation			Test Set		
	LAS	UAS	LS	LAS	UAS	LS
set-0	67.45	87.85	70.37	66.90	87.18	70.02
set-1	67.96	88.96	70.57	67.65	88.06	70.59
set-2	67.77	87.94	70.66	67.06	88.63	69.40
set-3	68.31	89.50	70.37	68.28	88.98	70.45

TABLE V
ACCURACIES ON THE COARSE-GRAINED DATASETS

Dataset	Cross-Validation			Test Set		
	LAS	UAS	LS	LAS	UAS	LS
set-0	71.87	87.97	75.37	69.32	88.91	72.10
set-1	73.24	89.55	75.78	71.80	90.14	74.22
set-2	72.20	88.32	75.33	69.73	88.29	72.07
set-3	73.61	89.86	75.89	71.29	89.98	73.29

on set-1. The accuracies obtained on this set reflect the effect of splitting just one type of external sandhi. This is understandable given that the number of sandhi splitting changes in the treebank is much less than the error corrections made to create set-1 (see table II). In the cross-validation experiments with set-1, we observed that the performance of the parser improves as the number of folds is increased. This suggests that the parser needs more training data to learn the new structures created as a result of sandhi splitting in the treebank.

The performance of the parser on set-3 is significantly better than both set-1 and set-2. The increase is significant in both LAS and UAS. This shows that post-editing changes such as POS-tag and chunk tag corrections as well as dependency corrections also aid in the learning of sandhi split structures. The improvement in UAS (1.65 for cross-validation and 1.80 on the test set) is more than that of LAS (0.86 for cross-validation and 1.38 on the test set).

We also repeated this experiment with similar datasets created using coarse-grained data [9]. The parsing accuracies on the coarse-grained datasets are shown in Table V. The results of this experiment exhibit a trend similar to that observed in the case of fine-grained data. However, it must be noted that the increase in LAS is expectedly much higher in the case of coarse-grained data. Overall, the results justify our claim about the importance of splitting sandhi forms in the treebank for the task of NL parsing.

VII. CONCLUSIONS

In this paper, we introduced the linguistic phenomenon of external sandhi in Telugu, an Indian language. We discuss how external sandhi formation in Telugu poses a problem in the syntactic annotation of Telugu sentences. We show using examples, that external sandhi, if not handled prior to dependency annotation in the treebanking process, can lead to either loss or misrepresentation of syntactic information. We report the insights gained from a detailed study of the instances of external sandhi from version 0.2 of the Telugu treebank. Based on these insights, we propose a modification to the generic annotation pipeline which would be relevant for all Sandhi languages. We manually split instances of one type of external sandhi widely occurring in the previous version of the treebank. In addition to sandhi-splitting, post-editing changes which include POS-tag corrections, chunk tag corrections and dependency (both attachment and label) corrections were also carried out, resulting in the development of a new version 0.5 of the Telugu treebank. Finally, we conduct an experiment with a statistical parser to empirically verify the usefulness of sandhi-splitting for the NL parsing task. The results of our experiment show that splitting instances of even just one type of external sandhi has a salubrious effect on the overall parsing accuracies. Developing an automatic sandhi-segmenter for Telugu based on our experience of manual sandhi-splitting is part of our immediate future work.

[9]The number of distinct dependency labels in the fine-grained data (44) is reduced to 22 coarse-grained labels.

ACKNOWLEDGMENTS

The authors would like to thank Viswanatha Naidu, Samar Husain, Prashanth Mannem, Sukhada and Sriram Venkatapathy for the helpful comments and discussions. Thanks are also due to all the annotators who did the initial annotation. The authors especially appreciate Viswanatha Naidu's sterling efforts that led to the development of HyDT-Telugu 0.2.

REFERENCES

[1] M. Marcus, M. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn Treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[2] A. Bharati, V. Chaitanya, and R. Sangal, *Natural language processing: a Paninian perspective*. Prentice Hall of India, 1995.

[3] J. Hajič, "Building a Syntactically Annotated Corpus: The Prague Dependency Treebank," *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*, 1998.

[4] E. Hajicová, "Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation," *Proceedings of TSD'98*, pp. 45–50, 1998.

[5] K. Oflazer, B. Say, D. Hakkani-Tür, and G. Tür, "Building a Turkish treebank," *Treebanks: Building and Using Parsed Corpora*, vol. 20, pp. 261–277, 2003.

[6] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 423.

[7] F. Reichartz, H. Korte, and G. Paass, "Dependency tree kernels for relation extraction from natural language text," *Machine Learning and Knowledge Discovery in Databases*, pp. 270–285, 2009.

[8] R. Begum, S. Husain, A. Dhwaj, D. Sharma, L. Bai, and R. Sangal, "Dependency annotation scheme for Indian languages," *Proceedings of IJCNLP-2008*, 2008.

[9] R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D. Sharma, and F. Xia, "A multi-representational and multi-layered treebank for hindi/urdu," in *Proceedings of the Third Linguistic Annotation Workshop*. Association for Computational Linguistics, 2009, pp. 186–189.

[10] M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D. Sharma, and F. Xia, "Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure," in *The 7th International Conference on Natural Language Processing*, 2009, pp. 14–17.

[11] A. Bhatia, R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D. Sharma, M. Tepper, A. Vaidya, and F. Xia, "Empty Categories in a Hindi Treebank," in *LREC-2010*, 2010.
[12] A. Bharati, M. Bhatia, V. Chaitanya, and R. Sangal, "Paninian Grammar Framework Applied to English," *South Asian Language Review*, 1997.
[13] A. Vaidya, S. Husain, P. Mannem, and D. Sharma, "A Karaka Based Annotation Scheme for English," *Computational Linguistics and Intelligent Text Processing*, pp. 41–52, 2009.
[14] C. Vempaty, V. Naidu, S. Husain, R. Kiran, L. Bai, D. Sharma, and R. Sangal, "Issues in Analyzing Telugu Sentences towards Building a Telugu Treebank," *Computational Linguistics and Intelligent Text Processing*, pp. 50–59, 2010.
[15] B. Krishnamurti, *The Dravidian languages*. Cambridge Univ Press, 2003.
[16] E. Sapir, *Language: An introduction to the study of speech*. Dover Publications, 1921.
[17] J. Greenberg, "A quantitative approach to the morphological typology of language," *International Journal of American Linguistics*, vol. 26, no. 3, pp. 178–194, 1960.
[18] S. Husain, "Dependency Parsers for Indian Languages," *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, 2009.
[19] A. Bharati, R. Sangal, D. Sharma, and L. Bai, "Annotating corpora guidelines for pos and chunk annotation for indian languages," 2006, technical report: TR-LTRC-31, LTRC.
[20] S. Buchholz and E. Marsi, "CoNLL-X shared task on multilingual dependency parsing," in *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2006, pp. 149–164.
[21] J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret, "The CoNLL 2007 shared task on dependency parsing," in *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*. Association for Computational Linguistics, 2007.
[22] V. Mittal, "Automatic Sanskrit segmentizer using finite state transducers," in *Proceedings of the ACL 2010 Student Research Workshop*. Association for Computational Linguistics, 2010, pp. 85–90.
[23] A. Bharati, D. Sharma, S. Husain, L. Bai, R. Begum, and R. Sangal, "AnnCorra: Treebanks for Indian languages, guidelines for annotating Hindi dependency treebank," 2009, http: // ltrc.iiit.ac.in / MachineTrans / research/tb/DS-guidelines/DS-guidelines-ver2-28-05-09.pdf.
[24] B. Ambati, S. Husain, J. Nivre, and R. Sangal, "On the role of morphosyntactic features in Hindi dependency parsing," in *The First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, 2010, pp. 94–102.
[25] A. A. Macdonell, *A Sanskrit Grammar for students*. New Delhi, India: D.K. Printworld (P) Ltd., 1926.
[26] E. Selkirk, "On prosodic structure and its relation to syntactic structure," *Nordic Prosody II: Papers from a Symposium*, pp. 111–140, 1981.
[27] A. Zwicky, "Stranded *to* and phonological phrasing in english," *Linguistics*, vol. 20, pp. 3–57, 1982.
[28] H. Andersen, *Sandhi phenomena in the languages of Europe*. Mouton de Gruyter, 1986.
[29] M. Absalom and J. Hajek, "Prosodic phonology and raddoppiamento sintattico: a re-evaluation," in *Selected Papers from the 2005 Conference of the Australian Linguistic Society, Melbourne: Monash University. http://www. arts. monash. edu. au/ling/als*, 2006.
[30] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "MaltParser: A language-independent system for data-driven dependency parsing," *Natural Language Engineering*, vol. 13, no. 02, pp. 95–135, 2007.

Keyword Identification within Greek URLs

Maria-Alexandra Vonitsanou, Lefteris Kozanidis, and Sofia Stamou

Abstract—**In this paper we propose a method that identifies and extracts keywords within URLs, focusing on the Greek Web and especially on URLs containing Greek terms. Although there are previous works on how to process Greek online content, none of them focuses on keyword identification within URLs of the Greek web domain. In addition, there are many known techniques for web page categorization based on URLs but, none addresses the case of URLs containing transliterated Greek terms. The proposed method integrates two components; a URL tokenizer that segments URL tokens into meaningful words and a Latin-to-Greek script transliteration engine that relies on a dictionary and a set of orthographic and syntactic rules for converting Latin verbalized word tokens into Greek terms. The experimental evaluation of our method against a sample of 1,000 Greek URLs reveals that it can be fruitfully exploited towards automatic keyword identification within Greek URLs.**

Index terms—**Greek to Latin character set transliteration, Greeklish to Greek transliteration, keyword extraction, Uniform Resource Locator, word segmentation.**

I. Introduction

THE large volume of information that is available over the web increases at prodigious rates with the current size of the surface web reaching to nearly 15 billion pages [1]. This, coupled with the need for accurate and effective identification of useful information within this huge network of data sources, has made imperative the need to come up with efficient methods for organizing, processing and structuring the plentiful web content. Towards this direction, several researchers have proposed methods for classifying the web content thematically so as to facilitate the data storage and the information seeking processes. The most commonly employed approaches towards web data classification focus on the examination of three main features extracted from web pages, namely their textual content [2], their anchor text and internal link distribution [3] and their URL features [4], [5], [6], [7], [8]. Although there exist several techniques for each of the above approaches, there is still ample room for improvements as none of the existing tools and methods can successfully detect the topic of every single page on the web and thus be able to assign it to a suitable thematic category.

Manuscript received November 1, 2010. Manuscript accepted for publication January21, 2011.

M.-A. Vonitsanou is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece (e-mail: bonitsan@ ceid.upatras.gr).

L.Kozanidis is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece (e-mail: kozanid@ ceid.upatras.gr).

S.Stamou is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece and the Department of Archives and Library Science, Ionian University, 49100, Greece (e-mail: stamou@ ceid.upatras.gr).

In this paper, we address the problem of URL-based keyword extraction for web page classification from the perspective of a Greek Web search engine. In particular, we study the problem of URL features analysis in order to capture web resource's thematic orientation. The extracted features can be used in order to automatically categorize Greek web content based entirely on URLs and without the need to perform content analysis which is a time-consuming and laborious process. The motive for carrying out our study is the observation that Greek URLs are articulated via the use of Latin characters and as such even if they encapsulate useful information within their elements, we have to translate this information to Greek in order to be able to interpret it.

In this paper, we address the problem of URL-based keyword extraction for web page classification from the perspective of a Greek Web search engine. In particular, we study the problem of URL features analysis in order to capture web resource's thematic orientation. The extracted features can be used in order to automatically categorize Greek web content based entirely on URLs and without the need to perform content analysis which is a time-consuming and laborious process. The motive for carrying out our study is the observation that Greek URLs are articulated via the use of Latin characters and as such even if they encapsulate useful information within their elements, we have to translate this information to Greek in order to be able to interpret it.

It is common knowledge that the URL is a string that specifies the mechanism to retrieve the identified sources, providing a scheme, a host or IP address and a path. Relying on pages' URLs rather than their textual content for web data classification is more time and cost effective since working with strings (i.e. URLs) instead of full documents diminishes the computational complexity and the network overheads associated with data processing. Extracting keywords from a URL can be useful because when no anchor text exists or the web resource is not a web page, it is the only available information about the web resource of interest. URL-based web page classification mainly concerns the identification of keywords within URLs that could serve as terminological descriptors of the corresponding pages' topics. But keyword extraction from URLs is not an obvious straight forward process, because URLs may or may not contain valid terms, they might contain symbols, special characters, they may conflate alphanumerics to abbreviate a phrase or a name and so forth. Conversely to the URLs' content, which is difficult to capture and interpret as this does not follow any specific guidelines, the URLs' structure is indicative of the topology of their corresponding web pages on the web graph. Therefore, the majority of works that try to capture the properties of web

pages based on the analysis of their URLs mainly focus on building URL parsers that could interpret the URL syntax. The few reported attempts that try to identify the topic of a web page based on the interpretation of the keywords identified with the page's URL, generally focus on English URLs. In this paper, we focus on the analysis of Greek URLs in order to identify and extract meaningful terms from their elements. The main challenge we need to confront is the fact that Greek URLs contain terms written in Latin and that Greek words can be transliterated in many different ways such as phonetic, orthographic or visual, depending on personal references. In addition, apart from the syntactically valid work combinations within URLs, the Greek language being a free word order one, allows multiple combinations of terms, some of which result to word phrases not necessarily encoded in general-purpose dictionaries.

Although there are previous works on how to process Greek online content [9], [10], [11] none of the reported attempts has focused on the problem of keyword identification within URLs of the Greek Web domain. In our work, we address the problem of keyword extraction from Greek URLs by implementing a system that integrates a transliteration engine and a URL tokenizer. In brief, the URL tokenizer segments an input URL into tokens which are given as input to the transliteration engine, which in turn produces all possible variations of the URLs' Greek tokens. For generating the transliterations, our engine relies on a Greek morphological dictionary, a Greek grammar and embodies a set of orthographic rules. After a brief introduction to relevant works, we describe in detail our Greek URL-based keyword extraction method, we discuss the results of a preliminary study we carried out and we sketch our plans for future work.

II. Related Work

There exist large volumes of works on URL processing for web page classification. Among the existing studies, researchers proposed methods for segmenting URLs into meaningful chunks to which one could add components, sequential and orthographic features for modeling salient patterns and rely on them for web data organization [6].

From a different perspective, researchers suggested ways for categorizing web pages based on URL elements, metadata descriptors and text extraction techniques via three-phase pipeline of word segmentation, abbreviation expansion and eventually classification [4]. A slightly different approach [5] employs a two-phase pipeline (e.g. URL word segmentation/ expansion and classification) for reducing the content of web data sources and be able to classify pages from academic hosts into the following predefined categories: course, faculty, project and student. More recently in [7] researchers proposed a machine learning technique for identifying the topical subject of a page based on its URL feature analysis. Feature identification within URLs entailed the combination of token and n-gram representation models. From a different viewpoint in [8] URL-based web page classification relies on language detection methods and the resulting classification is according to the pages' language rather than theme. Still, the case of Greek has not been investigated in any of the related works.

Related work falls also within the subject of Greek transliterations using the Latin alphabet for enabling Greek web content management. In this direction the work of [9], [10] focuses on the identification of query keywords from Greek web content and the subsequent handling of web queries verbalized in Latin characters. In [11] the authors study ways for classifying web sites of Greek vendors based on the identification of entities within their contextual elements. With respect to Greek transliterations of Latin-scripted texts, researchers mainly rely on the application of probabilistic models [12], spell-checking techniques [13] and regular expressions approaches [14] which they unify into common transliteration platforms. Despite the availability of such tools and methods none of them has been tailored to handle transliterations embedded within URLs in which lexical boundaries are absent and there is a lack of consensus with respect to what could or should a URL contain so as to reflect the content of its hosting page.

III. The Greek Web

This section provides a brief description of the Greek Web, the Greek language and the characteristics of Greek transliterations using Latin characters

A. The Greek Web

The main difficulty in defining the properties and characteristics of the Greek web arises from the fact that the exact limits of the Greek web are vague and imprecise. A naive approach would be that the Greek Web consists of the sites registered in the .gr top-level domain. This claim would lead to incorrect results as many Greek Web sites are hosted under the .net, .com, or .org top-level domains and reverse many sites in the *.gr* domain verbalize their Greek-oriented content in via the use of English [15]. In the course of our study, we define the Greek web as the web content written in Greek. Although we are aware of the fact that our definition of the Greek Web is incomplete, we rely on that in the course of this study essentially because our research objective is to identify valid keywords within Greek URLs rather than determine and capture the boundaries of the Greek Web.

B. The Greek Language

Like most Indo-European languages, Greek is highly inflected. The Greek alphabet consists of 24 letters each with a capital and a lowercase form plus an extra form for the letter s when used in the final position. Greek demonstrates a mixed syllable structure, permitting complex syllabic onsets, but very restricted codes.

Greek is a language distinguished by an extraordinarily rich vocabulary and a powerful compound-constructing ability. Another distinctive characteristic of the Greek language is its rich inflectional morphology which may deliver for a single lemma between 7 (for nouns) to 150 (for verbs) distinct

inflected forms. In addition, due to the existence of diphthongs and digraphs, spelling is significantly complicated.

Based on the above characteristics of the Greek language, we may naturally conclude that computationally processing Greek texts is a complex and laborious process that requires extensive linguistic knowledge and the availability of several resources such as dictionaries, grammars, sets of rules, corpora, etc.

C. Greeklish

The transliteration of Greek to Latin characters, a frequent practice on the web, has formed a hybrid language known as Greeklish.

Greeklish became widely known in the 1990's since not all operating systems and applications, especially web browsers, had support for the Greek character set. Nowadays, they are commonly used in blogs and forums because they are typed easily and users do not have to follow any orthographic rules.

Greeklish is not standardized, thus Greek words can be transliterated to Latin script in many different ways. Briefly, there are two generic types of transliterations, namely [16]:

a) Phonetic transliterations: based on how words are pronounced. For example «καλημέρα» meaning "*goodmorning*" is transliterated to "*kalimera*".
b) Orthographic transliterations: based on how the words are written. The aforementioned example is transliterated to "*kalhmera*".

Yet, there still exist quite a few variations in both orthographic and phonetic transliterations of certain Greek characters. For instance, the Greek letter *θ* (theta) may be written as *8, 9, 0, q, u* in the orthographic use of Greeklish and *th* in the phonetic use. What makes things more complicated is that oftentimes people switch between phonetic and orthographic transliterations, therefore increasing the heterogeneity of Greeklish writing.

D. Challenges

Given the variety of Greek to Latin characters' transliterations it becomes evident that being able to accurately reproduce Greek lemmas from Latin-scripted words becomes cumbersome and error-prone. This is not only due to multiple mappings that hold between the Greek and the Latin character sets (e.g. Greek diphthongs may match a single Latin character) but also due to the absence of specific guidelines about how Greek words are transliterated in Latin and vice versa. In addition, the lack of punctuation marks in the Latin alphabet imports an additional burden in the process of the identification of the correct Greek term as there exist many homonyms in the Greek vocabulary. For example the Greek term «γέρος» meaning "old man" and «γερός» meaning "strong man" produce the same transliteration "geros". Similar error-prone situations emerge in the case of homophones, i.e. words pronounced the same way, but spelled differently. In those cases, the word interaction should be considered.

IV. Methodology

Given the lack of a standard transliteration for Greeklish, it is extremely difficult to automatically process Greeklish data. Because of that, research on keyword extraction from URLs has not addressed the case of Greek. In this section we present in detail our method for identifying keywords within Greek URLs. In particular, we introduce the main components our method incorporates, namely the URL tokenizer and the Latin-to-Greek transliteration engine. Alongside, we introduce the resources upon which our proposed components operate and we demonstrate via examples the URL keyword identification process.

A. URL Tokenization

Based on the observation that a significant fraction of the URLs contain two or more word-tokens that are not delimited by non alphanumeric characters [5], it is obvious that the implementation of a tokenizer is required. Briefly, the tokenizer searches within the substring of the input token (i.e. URL), for meaningful keywords. To tackle tokenization for Greek URLs, we implemented two distinct yet complementary tokenizers, namely a surface keyword tokenizer and a hidden keyword tokenizer, both of which operate upon dictionary lookups.

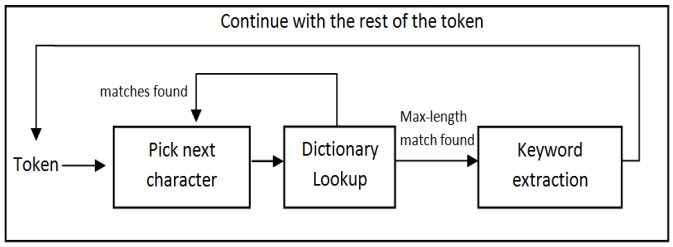


Fig. 1. Tokenization.

The surface keyword tokenizer parses the URL tokens from beginning to end, terminates upon the detection of an explicit keyword token. Unless the latter is identified, the tokenizer proceeds until reaching to a maximum size of keyword tokens and runs recursively by employing successive tokens as starting points simulating an n-gram examination process. For example, consider the URL http://www.contrastsensitivity.com where "contrastsensitivity" has to be split into "contrast" and "sensitivity". The tokenizer starting from position = 0 identifies "contrast" and starting from position = 8 identifies the keyword "sensitivity". The end of the token is identified, so the tokenizer terminates its function and returns the above keywords. This tokenizer works also when keywords are between unknown words. Figure 1 illustrates the tokenization process.

Consider now a typical property of URLs, i.e. that keywords are nested inside other keywords or unknown words. Obviously, using the aforementioned tokenization technique would not accurately identify the hidden URL keywords. To tackle this problem we implemented a hidden-keyword tokenizer, which begins from every character and searches for keywords continuously until reaching the end of the token. For

example, consider the word "certifications" *from which we want to extract the keyword "cat". The tokenizer first searches within "*certifications*", then moves to position = 1 and searches "*ertifications*", and so forth until it extracts all of the hidden keywords*.

B. Latin-to-Greek Transliteration Script Engine

The transliteration engine we implemented relies on recursive look-ups against a Greek dictionary and incorporates with a set of transliteration rules in order to effectively address the problem of the variety of Greeklish forms. In addition, to reduce the number of possible representations every Greeklish word might entail, we have integrated into our transliteration engine a set of grammatical and orthographical rules.

Based on Greeklish literature, large Greeklish corpora as well as our own experience, two different sets of transliteration rules are created, depending on whether the character is ambiguous or not. The dictionary is available in a trie structure in order to efficiently assume whether a substring is word-prefix or not.

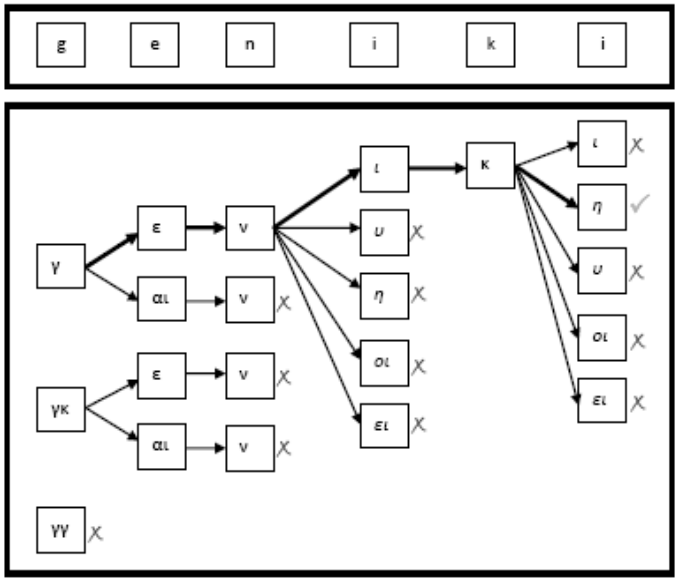


Fig. 2. Greeklish-to-Greek transliteration example.

The transliteration engine takes the following steps:

a) It replaces the unambiguous characters of a word with the corresponding characters of the Greek alphabet. For example: "geniki" (meaning general) is semi-translitered to "gενικι".
b) The output of the previous operation is processed letter by letter, in ascending order, in a way that a tree-like structure (Figure 2) is produced. Using the rules for ambiguous characters every letter is represented in all possible ways.
c) In every level (letter) a dictionary look-up is performed. If the given substring is word prefix, we move on to the next level. Otherwise, this branch is terminated and deleted.
d) As soon as the above process terminates, the output words are returned. In most cases only one word is returned.

A special issue that has to be addressed is the transliterations of two Latin characters to one Greek ("ph"→ "φ") and reverse ("ι"→ "οι"). Towards this direction we use a similar double-letter processing only on specific positions depending on whether the word contains such characters. Using the above method, it is obvious that computational burden is considerably reduced compared to methods that require the production of all the possible transliterations. Figure 2 schematically illustrates a transliteration example demonstrating the significant reduction of the required transliterations from 150 to 7. The complexity depends on the input string's length and the number of possible transliterations in every step.

Additionally, orthographic rules are also applied in order to extract misspelled keywords, like trying with double consonants. For example, gramata (meaning letters) is also tried as grammata in order to extract its correct form (γράμματα). In that way, we avoid the use of a Speller. Moreover, in order to create a Greeklish-to-Greek dictionary, containing usual URL keywords, we store locally every successfully one-way transliterated word. Thus, before applying the above method, a word is searched in the transliteration dictionary and the computational burden is reduced further.

C. URL Keywords' Extraction

Having presented the functionality of every sub-system that our URL keyword identification module integrates, we now proceed with the description of the keyword identification process. The keyword extraction system consists of the following steps:

a) A URL is divided into its basic components, according to URI protocol (scheme:// host / path-elements / document . extension).
b) The host-domain part is split on the appearance of punctuation marks.
c) For each token, transliteration is applied. Firstly a look up in the produced Greeklish-to-Greek dictionary is performed and if the word is not found, the transliteration machine is activated.
d) Parallel transliteration, tokenizing is performed. Every exact match is returned and the process continues to the next character.

Figure 3 schematically illustrates the URL keyword identification process.

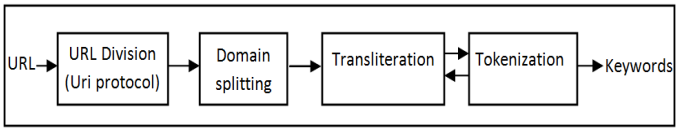


Fig. 3. URL's Keywords identification process.

TABLE I
EXAMPLES OF SPECIAL CASES

URL	Keywords Extracted	Correct Keywords	Explanation
http://www.pamediakopes.gr/	πάμε, διακοπές	πάμε, διακοπές	Two-keyword URL
http://www.politis-chios.gr/	πολίτης, πωλητής, Χίος	πολίτης, Χίος	Ambiguous Greeklish term
http://www.mila-elefthera.gr/	μήλα, μίλα, ελεύθερα	μίλα ελεύθερα	Homophones
http://www.ellinikospiti.gr	ελληνικός	ελληνικό, σπίτι	Inflection
http://www.iatridis.gr/	ιατροί	Ιατρίδης	Named entity
http://www.gatospito.com/	γάτος	γατόσπιτο	Unknown compound
http://www. skeftomastellinika.com/	σκεφτόμαστε, νίκα	σκεφτόμαστε, ελληνικά	Word Overlapping
http://www.tapote.gr/	ποτέ	ΤΑΠΟΤΕ	Unknown abbreviation

V. EXPERIMENTAL EVALUATION

In this section we outline a preliminary experimental study we carried out in order to assess the performance of our method in accurately identifying meaningful keywords within Greek URLs.

A. Experimental Setup

To assess our study objective, we applied our proposed technique on a set of sample URLs. This dataset was collected manually and consists of 1,000 distinct URLs, which belong to different domain names, containing Greek terms. Via the use of a general purpose web crawler we downloaded every web page corresponding to the above URLs and we extracted its title and the meta-keywords, i.e. keywords that the webpages' authors have identified for describing the respective pages' content and are found as values to html meta-tags. From the initial dataset, we could only download 958 web pages of which only 942 had an associated title, 547 contained meta-words and only 400 of them had both an associated title and meta-words. In cases that the page title or meta-words were missing the only information we had at our disposal besides the page content was the page URL. For our experiments we used only the 400 URLs that contained both title and meta-words.

We compared the title and meta-words extracted against the keywords our method identified in the corresponding URLs. To assess our method's effectiveness in detecting valid keywords within Greek URLs we carried out two experiments:

a) One using the surface keyword tokenizer, and
b) One using the hidden-keyword-tokenizer.

B. Experimental Results

We perform exact match assessment; each extracted keyword is searched in the title or meta-keywords of the web page.

Using the surface keyword tokenizer, 49% of the extracted keywords per URL were found in the URL's title, 43% in the meta-words, 33% in both title and meta-words and 60% in title or meta-words. Using the hidden-keyword-tokenizer, the results reduced significantly, due to the large number of keywords extracted per URL.

The main weakness of our proposed system it that when named entities are contained within Greek URLs, it fails to recognize them as such and therefore it is ineffective in extracting keywords from them. As a consequence, obtained results might be misleading especially when dealing with named entities not lexicalized in the dictionary. In addition, several web sites contain in their titles or meta-keywords terms such as "Home", "Introduction", the domain itself or non-Greek words, instead of containing topic keywords.

Moreover, we have to consider that Greek words are highly inflected. Thus, as every word might occur in many different forms, the exact matching would not recognize the word similarity, and the results can be misleading. In light of this observation, a Greek lemmatizer or stemmer should be incorporated in the comparison task.

Nevertheless, despite the above few error-prone situations, results demonstrate that in overall, our proposed technique can effectively capture a considerable amount of valid keywords within URLs. This coupled with the acknowledged lack of existing keyword detection techniques from Greek URLs validates the usefulness and the potential of our proposed method towards organizing Greek web content based entirely on the analysis of their URLs.

VI. CONCLUSION

In this paper we introduced a keyword extraction technique focusing on Greek URLs. Our proposed technique consists of two main subsystems: a transliteration engine and a tokenizer. The transliteration engine produces the possible reconstructions of the Greeklish tokens using a dictionary in order to reduce them. After transliteration, if a token consists

of more than one keyword, the tokenizer segments the Greek tokens into meaningful keywords. Based on our experimental results, this paper shows that quality keyword extractors for Greek web pages can be built based on URLs alone.

The innovative aspect of our work is that we process non-English URLs, particularly URLs that contain keywords written in Greek using Latin characters.

Future work concentrates on, but is not limited to the following issues. Abbreviation handling is a significant issue in URL processing. For this study we used a common-abbreviations list. However we are working on an abbreviations' identifier based on Greek URLs. In addition, like previous attempts, we will process the path and the query part of the URL, adding a system that recognizes and decompiles percent encoding. Moreover, we are planning to improve the hidden-keyword tokenizer in order to reduce the extracted keywords that are not related to the URL. We are also working on adding a Greek stemmer to each extracted keyword and obtain keywords synonyms using Wordnet [16] in order to receive an improved match between URL keywords and meta- or title- keywords. Finally, the experiments will be repeated using a larger data set and a language detector, in order to recognize English keywords that may be contained within a Greek URL.

REFERENCES

[1] The size of the World Wide Web. Available: http://www.worldwidewebsize.com.

[2] S. Dumays and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd annual international ACM SIGIR Conference on Research and development in information retrieval*, Velingrad, Bulgaria, 2000, pp. 256–263.

[3] S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in *Proceedings of the International World Wide Web Conference (WWW2002)*, Honolulu, 2002, pp. 251–262.

[4] M.-Y. Kan, "*Metadata extraction and text categorization using Universal Resource Locator expansions*", National University of Singapore, Department of Computer Science, Technical Report, TR 10/03, 2003.

[5] M.-Y. Kan, "Web page classification without the web page," in *Proceedings of the 13 th. International World Wide Web Conference (WWW2004)*, New York, USA, 2004, pp. 262–263.

[6] M.-Y. Kan and H.-O.-N. Thi, "Fast webpage classification using url features," in *CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management.* New York, USA: ACM, 2004, p. 325-326.

[7] E. Baykan, M. Henzigner, L. Marian, and I. Weber, "Purely url-based topic classification," in *Proceedings of the 18th international World Wide Web Conference (WWW2009)*, Madrid, Spain, 2009, pp. 1109–1110.

[8] E. Baykan, M. Henzigner, and I. Weber, "Web page language identification based on URLs," in *Proceedings of the VLDB Endowment 1(1)*, Auckland, New Zealand, 2008, pp. 176–188.

[9] S. Stamou, L. Kozanidis, P. Tzekou, and N. Zotos, "Query selection for improved Greek web searches." in *Proceedings of the 2nd International CIKM Workshop on Improving Web Retrieval for non-English Queries*, CA, USA, 2008, pp. 63–70.

[10] P. Tzekou, S. Stamou, N. Zotos, and L. Kozanidis, "Querying the Greek web in greek-lish," in *Proceedings of the SIGIR Workshop on Improving Web Retrieval for non-English Queries*, Amsterdam, Netherlands, 2007, pp. 29–38.

[11] D. Farmakiotou, V. Karkaletsis, G. Samaritakis, G. Petasis, and D. Spyropoulos, "Named entity recognition in Greek web pages," in *Proceedings Companion Volume of 2nd Hellenic Conference on Artificial Intelligence (SETN-02)*, Thessaloniki, Greece, 2002, pp. 91–102.

[12] A. Chalamandaris, A. Protopapas, P. Tsiakoulis, and S. Raptis, "All greek to me! an automatic greeklish to greek transliteration system," in *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, 2006, pp. 1226–1229.

[13] Aspell, spell checker for Greek. Available: http://aspel.source.gr.

[14] A. Karakos, "Greeklish: An experimental interface for automatic transliteration," *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 1069–1074, 2003.

[15] C. Lampos, M. Eirinaki, D. Jevtuchova, and M. Varzigiannis, "Archiving the greek web," in *Proceedings of the 4th Intl. Web Archiving Workshop*, Bath, UK, 2004.

[16] WordNet. Available: http://www.cogsci.princeton.edu/~wn.

Contextual Analysis of Mathematical Expressions for Advanced Mathematical Search

Keisuke Yokoi, Minh-Quoc Nghiem, Yuichiroh Matsubayashi, and Akiko Aizawa

***Abstract*—We found a way to use mathematical search to provide better navigation for reading papers on computers. Since the superficial information of mathematical expressions is ambiguous, considering not only mathematical expressions but also the texts around them is necessary. We present how to extract a natural language description, such as variable names or function definitions that refer to mathematical expressions with various experimental results. We first define an extraction task and constructed a reference dataset of 100 Japanese scientific papers by hand. We then propose the use of two methods, pattern matching and machine learning based ones for the extraction task. The effectiveness of the proposed methods is shown through experiments by using the reference set.**

***Index Terms*—Natural language processing, mathematical expressions, pattern matching, machine learning.**

I. Introduction

MATHEMATICAL expressions often play an essential part in scientific communications. It is not only that they are used for numerical calculations, but that they are used for conveying scientific knowledge with less ambiguity, enabling researchers to precisely define and formalize target problems. They are also used for proving the validity of newly discovered properties. Facilitating cross-document retrieval of mathematical expressions encourages better understanding of the content: what a formula means, why it was used there, or how it was derived. However, regardless of the importance in knowledge-oriented information access, there have been only a few studies on mathematical searches so far. Consequently, with current search engines, most of the mathematical expressions are either totally excluded from the search or only a fraction of those mathematical symbols are indexed and retrieved.

Our purpose is to propose a new framework for a mathematical content search based on semantic analysis of the content. As mathematical expressions are highly abstracted and hard to manage without the accompanying natural language text, we utilize both the structure of expressions and natural language descriptions surrounding them (Fig. 1). It should be noted here, that the existing few studies on mathematical search relied solely on notation similarity of equations and do not use any context information. As far as we know, our research is a first practical attempt to use both the structure of mathematical expressions and the related descriptions within the same framework. We focus initially on a technique for connecting *elements* of mathematical expressions with their names, definitions and explanations, which we collectively call *mathematical mentions*. Examples of elements in this case are variables, functions, or other components that correspond to some newly introduced mathematical concepts in a target document.

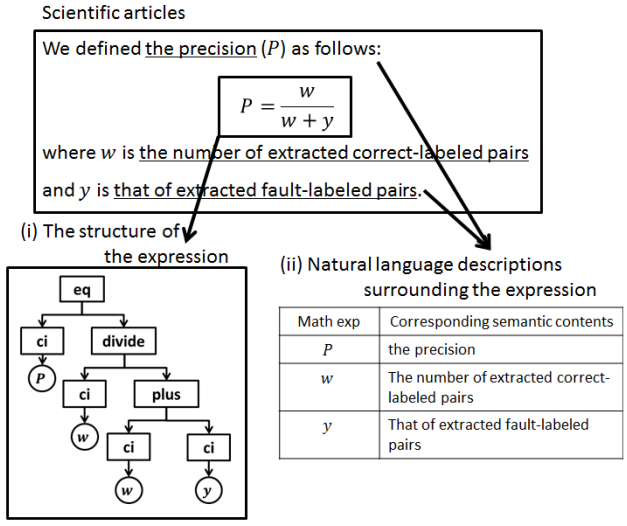


Fig. 1. Illustrative example of proposed mathematical content search.

As a target dataset, we selected 100 scientific papers in computer science published by the Information Processing Society of Japan [1]. First, all the mathematical expressions contained in the dataset were converted into Mathematical Markup Language (MathML) format, initially using Math OCR software and then by human check for validating and correcting unavoidable mistakes. Here, MathML is a common standard format for mathematical expressions. All the names and definitions with explicit reference to any of the MathML elements were then also manually annotated. For example, given a statement "Let e be the base of natural logarithm",

Manuscript received November 12, 2010. Manuscript accepted for publication January 10, 2011.

Keisuke Yokoi is with Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan (e-mail:kei-yoko@nii.ac.jp).

Minh-Quoc Nghiem is with Department of Informatics, The Graduate University for Advanced Studies, Tokyo, Japan (e-mail:nqminh@nii.ac.jp).

Yuichiroh Matsubayashi is with National Institute of Informatics, Tokyo, Japan (e-mail:y-matsu@nii.ac.jp).

Akiko Aizawa is with Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan and with National Institute of Informatics, Tokyo, Japan (e-mail:aizawa@nii.ac.jp).

the phrase "the base of natural logarithm" is annotated as a referrer to the mathematical element "e".

The task we define in this paper is to automatically identify the referrer/referee pairs on the above target dataset. As the majority of mathematics-related descriptions follow a limited number of template expressions, we apply a supervised machine learning framework in our approach. First, frequently appearing description patterns are collected from a separately prepared reference set. Next, using the basic patterns and other linguistic information as features, a support vector machine (SVM) is trained to decide whether given candidate pairs correspond to each other or not. The effectiveness of the proposed method is investigated using the annotated data in our experiments. It is better than the method using pattern matching.

The contribution of our paper is as follows: First, we show the importance of semantic mathematical search and introduce a new framework for extending the current mathematical search systems. For this purpose, we propose the use of a machine learning-based method that identifies the correspondence between mathematical expressions and their natural language descriptions. Second, we manually construct an annotated corpus and evaluate the performance of our method. We show that a supervised machine-learning framework can be used effectively with about 87% precision and 81% recall. Third, we define a new type of information extraction task to identify equivalent relations between natural and formal languages. Our investigation shows that our framework had a satisfactory performance for this type of problem with technical writings.

II. Related Works

We assumed mathematical expressions are represented using Mathematical Markup Language (MathML) [2]. Although it is not widespread, MathML is a worldwide standard defined for mathematical expressions recommended by W3C [3], and as such, is supported by many existing Web browsers. An increasing number of MathML compatible software tools have become available, including editors, mathematics software packages, and translators between MathML and other representations such as TEX or OpenMath; there is also Math OCR software to recognize mathematical expressions printed on paper [4].

Several researchers have done mathematical searches by using MathML and other formal languages for mathematics. Their research can be categorized by their primary goals, *mathematical search* and *mathematical knowledge-base*.

Research on *mathematical search* targets retrieving real-world mathematical documents in digital libraries or on the Web. Since such documents have a great deal of semantic ambiguity, the majority of mathematical search systems calculate similarity between mathematical concepts by considering syntactic information of the formulas. Munavalli et al. analyzed mathematical expressions written in MathML and translated the feature elements into index terms in their MathFind search system [5]. Mišutka et al. also extended the full text search engine with a formula tokenizer that converts formulas into representations of different generalized levels [6]. Adeel et al. generated keywords by using regular expressions for the mathematical equations written in MathML, and threw them to existing search systems as queries [7]. And we also proposed the use of a method for doing a similarity search for mathematical equations based on a distance calculation defined for the tree structure of MathML [8]. On the other hand, research on *mathematical knowledge-base* aims to automatically construct a comprehensive knowledge, or ontology, of mathematics. Therefore, these researches center in extracting rules or relations between mathematical elements from mathematics textbooks or documents. Kohlhase et al. proposed the use of a web-based, distributed mathematical knowledge base where relations between mathematical objects such as symbols, definitions, or proofs were stored in a database and utilized as mathematical facts [9]. Jeschke et al. presented a framework for automatic extraction of mathematical ontology from mathematical texts using natural language processing [10]. Although their framework is remarkable, general, and applicable to many mathematics systems, syntactic analysis of mathematical expressions was still left for future study.

To summarize, existing *mathematical search* studies mainly worked on "syntactic" information of mathematical formulas to identify mathematical concepts useful for indexing. Contrarily, most *mathematical knowledge-base* studies focused on the "semantic" information to extract relations between mathematics related entities. However, "syntactic" disambiguation of mathematical expressions often requires "semantic" interpretation; for example, deciding whether a symbol in an equation is a variable or a function without context information is sometimes difficult. Conversely, "semantic" information alone is often insufficient to identify precise mathematical relationship between the target elements. The final goal of our research is to combine both of the syntactic and semantic features to enable deeper analysis of mathematical expressions. For this purpose, we dedicate ourselves to extracting correspondence between mathematical elements and natural language descriptions.

III. Dataset Construction

Since no annotated corpus is available for MathML documents, we first constructed a dataset that we can use to develop and evaluate our method. The flowchart of the construction is shown in Fig. 2.

First, in the selection phase, we chose 214 papers related to the machine-learning field using a keyword list shown in Table I. We then removed 52 papers with only few mathematical expressions (162 candidates remained), and narrowed the candidate again in terms of relationship with each other, in particular, in the reference network (104 candidates remained) because this is the first step therefore it is desirable that target papers are relative as far as possible. Since we plan to extend

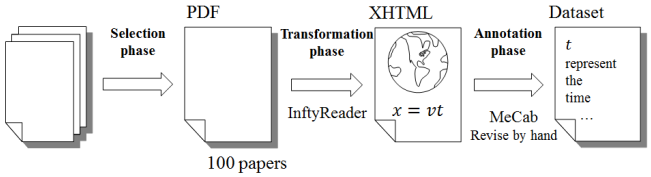


Fig. 2. Flowchart for dataset construction.

our work to mathematical content search in future, we intended our corpus to focus on a specific research topic so that the papers stand on some common mathematical grounding. We expect that with varied authors and years of the publications, sufficient diversity is still maintained for natural language expressions in the corpus.

TABLE I
KEYWORDS USED TO COLLECT RELATIVE (MACHINE LEARNING-RELATED) PAPERS

No.	Keywords (Japanese)	Keywords (English)
1	機械学習	Machine learning
2	教師あり学習	Supervised learning
3	教師なし学習	Unsupervised learning
4	サポートベクタマシン (SVM)	Support vector machine (SVM)
5	ニューラルネットワーク	Neural network

In the transformation phase, the 104 PDF papers were transformed into XHTML format where a mathematical OCR software, InftyReader [4], was used to convert printed mathematical expressions into MathML representations with manual consistency check.

In the annotation phase, we manually enumerated all the pairs of mathematical expressions and corresponding mathematical mentions. First we normalized each sentence (e.g. remove HTML tags) and then split it in morphemes by using a Japanese language morphological analyzer MeCab [11] and then put BIO tags on them to show whether each word correspond to each mathematical expression. For simplication, we only considered compound nouns as candidates for mathematical mentions here. Although mathematical mentions are often expressed as complicate noun phrases with prepositions, adjectives, or adverbs, we annotated only the last compound nouns in the phrases (note that Japanese language is a head-final language). After this process, the four papers without any pairs of mathematical expressions and descriptions were removed from the corpus, which resulted in 100 annotated papers left.

An example sentence in this dataset is shown in Table II. The target sentence can be translated into English as “Here, distribution Exp1 represents the prior probability distribution of the parameter Exp2” where Exp1 and Exp2 represent mathematical expressions. Each target expression is labeled independently using a separated column. In this case, Exp1 has two corresponding mathematical mentions “distribution (分布)” and “the prior probability distribution (事前確率分布)” and therefore these words are put B or I tags. Since only noun phrases are considered as candidates of mathematical mentions in our framework, B/I tags are not put on the phrase “the prior probability distribution of the parameter Exp2” but instead on “the prior probability distribution” as the second mathematical mention of Exp1.

TABLE II
EXAMPLE SENTENCE IN THE DATASET

ID	Morpheme	Tags	
0	ここ (here)	O	O
1	で	O	O
2	、	O	O
3	分布 (distribution)	B	O
4	Exp1	Pred	O
5	は	O	O
6	パラメータ (parameter)	O	B
7	Exp2	O	Pred
8	の	O	O
9	事前 (prior)	B	O
10	確率 (probability)	I	O
11	分布 (distribution)	I	O
12	を	O	O
13	示す (represent)	O	O

IV. METHODS FOR IDENTIFYING CORRESPONDING DESCRIPTION

In this section, we propose the use of two methods for identifying mathematical mentions corresponding to each mathematical expression: pattern matching and one based on machine-learning.

A. Basic Approach

Given a target mathematical expression, the objective here is to find phrases that represent a meaning, definition, or name of the expression. Multiple phrases can be the correct mathematical mentions for a certain mathematical expression. To simplify the problem, we presuppose that: first, all of the mathematical mentions are nouns or compound nouns and second, these mentions co-occur with the target mathematical expression within the same sentence. The problem is then attributed to the binary categorization of each noun phrase in the same sentence with the target mathematical expression.

Our basic approach for this task consists of two steps. First, the sentence containing a target mathematical expression is parsed by a morphological analyzer and the noun phrases are extracted using simple extraction rules; continuous nouns are combined to form a compound noun. Second, for each noun phrase in the sentence, a binary classification is applied to decide whether the phrase is a corresponding mathematical mention to the target or not. Note that each noun phrase in the sentence is processed multiple times if the sentence contains several mathematical expressions. If we take a sentence in Table II as an example, we see that the sentence includes two mathematical expressions (Exp1 and Exp2) and four noun phrases (“here (ここ)”, “distribution (分布)”, “parameter (パラメータ)”, “the prior probability distribution (事前確率分布)”). We, therefore, obtain eight candidate instances to classify

([Exp1, “here”], [Exp1, “distribution”], [Exp1, “parameter”], [Exp1, “the prior probability distribution”], [Exp2, “here”], [Exp2, “distribution”], [Exp2, “parameter”], and [Exp2, “the prior probability distribution”]) in total (Fig. 3).

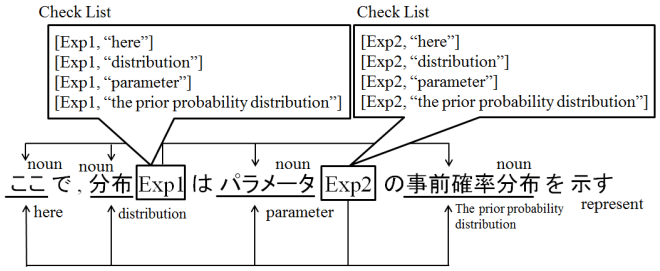


Fig. 3. Classification candidates of a sentence.

B. Pattern Matching Based Method

Our first attempt is based on a naive assumption that scientific papers use a limited number of template expressions to describe the meanings of mathematical expressions. The method based on pattern matching is used to clarify how well the mathematical mentions can be obtained by using a few representative patterns between a mathematical expression and a mathematical mention. We extract most frequent eight patterns from five randomly selected papers in IPSJ Journal (a journal on the field of information science) by hand. Note that these papers are not included in the dataset described in section III, where we choose only literature on machine learning. To keep a generality of the patterns, we did not restrict the topic of the papers. The extracted patterns are shown in Table III.

TABLE III
MOST FREQUENT EIGHT PATTERNS EXTRACTED FROM THE FIVE PAPERS

No.	Patterns
1	[Noun](+[AnotherExp]+“、”+...)+[Exp]
2	[Noun]+“を”(+...)+[Exp]+“と”+⟨する/表す⟩
3	[Exp]+“を”(+...)+[Noun]+“と”+⟨する⟩
4	[Exp]+“は”(+...)+[Noun]+“で”+⟨ある⟩
5	[Noun]+“と”+“呼び”(+...)+[Exp]+“で”+“表す”
6	[Noun]+“を/は”+[Exp]+“、”
7	[Exp]+“を/は”+[Noun]+“、”
8	[Exp]+“は”(+...)+[Noun]+“を”+⟨示す⟩

Here, [Noun] is a candidate noun, [Exp] is a mathematical expression, A function “$\langle v \rangle$” returns the root form of the verb v, the operator “/” denotes the or function, and “(+...)” indicates that there are zero or more words there. “(+ [AnotherExp] + “,” + ...)” indicates that there are zero or more sequences of another expression and comma. For instance, pattern 1 expresses the case that the [Noun] is the previous word of target [Exp] or the case that there are only some mathematical expressions and commas between the [Exp] and [Noun]. In Fig. 3, both [Exp1, “distribution”] and [Exp2, “parameter”] match pattern 1. The pair [Exp1, “prior probability distribution”] matches pattern 8.

Using these patterns, identification is performed as follows; given a pair of a mathematical expression and a candidate noun, a classifier returns *true* if the pair matches any of the patterns used and *false* if it does not. As a preliminary experiment, we confirmed that a classifier using above eight patterns achieved 85% in F-measure for another five randomly selected papers in the IPSJ Journal. We will evaluate the patterns in a larger dataset in section V.

C. Machine Learning Based Method

We also investigated a supervised learning approach to the task, using the basic patterns above and other linguistic information as clues for classification. As described in subsection IV-A, we formalized a problem as a binary classification for each noun phrase on the condition that the target mathematical expression and automatic morphological analysis are given. Here, we used an SVM-based binary classification model. The features that we used for the classification are shown in Table IV. Every feature in the table is expressed by using a binary value. The features are categorized into four types. First, the eight patterns extracted in the previous subsection are directly used as features. Second, several types of tokens which decide the structures of the sentence are used as clues for determining the relationship between [Noun] and [Exp]. Checking through the tokens between [Noun] and [Exp], this type of features tests the existence of commas and brackets, which decide the syntactic structures, and case markers of subject and object (“は” and “を”), which determine the argument structures between the [Noun] and [Exp]. Intuitively, the likelihood of the relationship between [Noun] and [Exp] may be lower if these features are triggered. Third, neighbor tokens of [Noun] and [Exp] are used as clues. And the last type feature is about dependency analyses. The dependency relation between the [Noun] and [Exp] must provide important clues for determining corresponding pairs.

Using a training set in section III, the L2-regularized L1-loss function is minimized with the Primal Estimated sub-GrAdient SOlver (Pegasos) algorithm [12]. We used the Classias [13] to estimate the parameters.

V. EXPERIMENTS AND DISCUSSIONS

This section gives experiments for evaluating each identification method. We divided the dataset described in section III into three subsets: 60 papers for training, 20 for development and 20 for testing. The training set has $3,867$ positive and $53,153$ negative instances, the development set has $1,267$ positive and $17,440$ negative instances, and the test set has $1,193$ positive and $16,219$ negative instances. We evaluate each model in terms of precision, recall, and F1-measure on the test set. We do not use the training and development set for the method based on pattern matching.

To make a baseline, we used a simple method that returns true iff the target noun phrase is the previous token of the

TABLE IV
FEATURES USED FOR MACHINE LEARNING

Features	Explanations
Pattern (1-8)	Are triggered if target pair matches each of eight patterns.
Another mathematical expression, comma, or opening/closing brackets	Test existance of another mathematical expression, comma, or opening / closing brackets between the target noun and the mathematical expression.
Case markers “は”, “を”	Test the existance of case marker “は” or “を” between taret noun and mathematical expression.
Other tokens	Test whether other types of tokens are clipped by targets or not.
Order	Test whether the target noun lies anterior to mathematical expression or not.
Noun	[Noun] itself.
Composition	Triggered if target noun is a compound noun.
Position from [Exp]	Integer numbers indicating a position from [Exp] (..., -2, -1, 1, 2, ...).
Previous/next words of [Noun]	Surface and PoS of previous/next word of target noun.
Previous/next words of [Exp]	Surface and PoS of previous/next word of target mathematical expression.
Nearest verb lemma	Lemma form of verb which first appears after latter target.
Word combination	Combination features using two to six features from features about near words.
Existence combination	Combination features using two to three features from features about existence.
Dependency relation	Tests whether the clause including target noun is dependent/head of that including mathematical expression / both clause have common head.

target mathematical expression. For example in Fig. 3, the baseline outputs two pairs [Exp1, “distribution (分布)”], [Exp2, “parameter (パラメータ)”].

The result of each model is shown in Table V and Table VI. We evaluated the following four models for the machine learning method: without pattern and dependency relation features (*w/o pat&dep*), without pattern features (*w/o pattern*), without dependency relation features (*w/o depend*), and with all the features (*All features*). We use two different evaluation criterions: based on soft and strict matching, respectively. With soft matching based evaluation, automatically extracted noun phrases are considered as *true* if they partially match human annotated ones. On the other hand, with strict matching based evaluation, the extracted phrases are considered as *true* only if they exactly agree with human annotated ones. While the former allows misidentification of the boundaries of the target noun phrases, the latter requires the exact identification.

The highest performance was achieved by machine learning model with dependency analysis features. Essentially, machine learning based models obtain higher precisions and much higher recalls than the method based on pattern matching. It can be seen that *w/o pattern* and *All features* models have no significant difference, which means the existence of pattern features doesn’t have much influence on the performance. These results suggest that the manually selected patterns

TABLE V
PRECISION/RECALL/F1-MEASURE OF EACH METHOD
(SOFT MATCHING EVALUATION)

Methods		Development set			Test set		
		Prec.	Recall	F1	Prec.	Recall	F1
Baseline		95.04	57.46	71.62	95.21	61.61	74.81
Pattern Matching		89.46	69.69	78.35	91.80	75.11	82.62
Machine Learning	w/o pat&dep	92.53	82.16	87.04	92.50	85.83	89.04
	w/o depend	92.46	82.24	87.05	92.59	85.83	89.06
	w/o pattern	92.21	83.11	87.42	92.45	86.17	89.20
	All features	92.20	83.03	87.38	92.45	86.17	89.20

TABLE VI
PRECISION/RECALL/F1-MEASURE OF EACH METHOD
(STRICT MATCHING EVALUATION)

Methods		Development set			Test set		
		Prec.	Recall	F1	Prec.	Recall	F1
Baseline		87.99	53.20	66.31	89.90	58.17	70.64
Pattern Matching		82.98	64.64	72.67	87.09	71.25	78.38
Machine Learning	w/o pat&dep	86.13	76.48	81.02	87.35	81.06	84.09
	w/o depend	86.07	76.56	81.04	87.43	81.06	84.12
	w/o pattern	85.60	77.43	81.45	87.32	81.39	84.25
	All features	85.89	77.35	81.40	87.32	81.39	84.25

were implicitly complemented by the combination of features obtained via SVM learning. On the other hand, the dependency feature contributed to the performance improvement. It can be presumed that dependency information successfully captured grammatically generalized and structural patterns which cannot be represented by using sequential patterns.

Note that the performance of the proposed method is upper-limited due to our preprocessing policy of compounds and multi-words described in subsection IV-A. It caused about 6% decrease in the overall performance.

As shown in Table VII, we also evaluated each pattern individually. Pattern 1 shows the best performance and the highest frequency while pattern 5 and 8 scarcely appeared in the dataset. The high precisions of the patterns 1, 2, and 6 exemplify that we extracted the set phrases by using these patterns. On the other hand, the result of pattern 5 and 8 suggests that patterns used for describing the meaning of mathematical expressions may vary depending on the topic/field of the paper. Eventually, we used the model with all these patterns, which achieved the highest performance in the development set as the best pattern model based on matching.

TABLE VII
RESULT OF EACH PATTERN

	Development set			Test set		
No.	Precision	Recall	F1	Precision	Recall	F1
1	93.70	58.72	72.20	94.07	65.13	76.97
2	87.50	1.66	3.25	95.35	3.44	6.63
3	52.94	1.42	2.77	30.00	0.50	0.99
4	76.40	5.37	10.03	80.36	3.77	7.21
5	NaN	0.00	NaN	0.00	0.00	0.00
6	78.57	1.74	3.40	100.00	1.76	3.46
7	55.56	0.79	1.56	66.67	0.50	1.00
8	NaN	0.00	NaN	NaN	0.00	NaN
All	89.46	69.69	78.35	91.80	75.11	82.61

In addition, to evaluate the sufficiency of the dataset, we plot the learning curve of the machine-learning model using all features in Fig. 4. The dataset is comparatively sufficient for machine learning, but, even if we use the maximum size of the training set, the curve still does not converge. Therefore, the gap between pattern matching and methods based on machine learning may increase, with the size of the dataset.

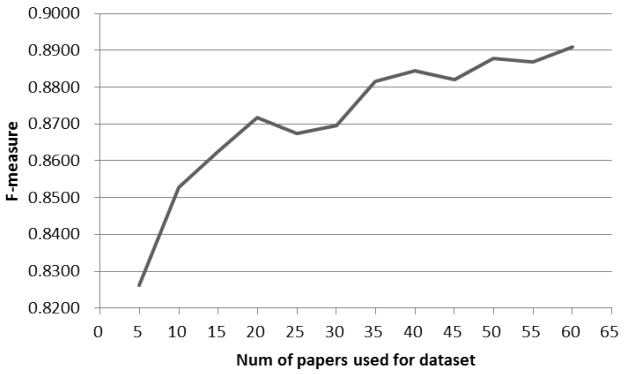


Fig. 4. Learning curve for the result using machine learning based method.

VI. Conclusion

We proposed the use of a method for extracting natural language descriptions associated with mathematical expressions in scientific papers. Our experimental results showed that the proposed machine learning framework works effectively with our dataset. We expect the performance can be further improved by using other information like mathematical expressions' structures. Since this is our first challenge for the mathematical search that includes both the syntactic and semantic aspects, in this paper we only focused here on the information extraction techniques to identify relationships between the two. We plan to incorporate the extracted information into the mathematical search system we already developed and to investigate the potential of the enhancement.

The remaining two important issues are constructing a dataset and determining mathematical mentions. First, the quality of datasets needs to be improved to enable more reliable evaluations. Our validation study showed the limitation of manual annotation particularly for appositions that frequently occur in a target dataset. For example, given a sentence like "*Distribution F is a prior probability distribution*", the apposition "*distribution F*" tends to be overlooked by a human annotator while automatic extraction methods evaluate this more accurately. Such an analysis suggests that the quality of the dataset can be improved by collecting candidates from different competing extraction methods and also by carefully reviewing. Second, we assumed that mathematical mentions are ones of the noun phrases in the same sentences as the target mathematical expressions. However, in real applications, other related descriptions are also useful. For example, given a sentence like "*W is a weight that controls the relative importance of the two operation points*", not only the term "weight" but also the succeeding that-clause is informative for users. This makes the determination of mathematical mentions a more challenging task and requires a reconfiguration of our task and dataset.

Finally, we expect the proposed scheme will be applicable to other languages as well because of the general tendency of mathematical descriptions to follow their characteristic patterns. They will be also addressed in our future study.

References

[1] "Information Processing Society of Japan," http://www.jpsj.or.jp/.
[2] World Wide Web Consortium, "Mathematical markup language (mathml) version 2.0 (second edition)," www.w3.org/TR/MathML2.
[3] "World Wide Web consortium (W3C)," http://www.w3.org/.
[4] M. Suzuki, T. Kanahori, N. Ohtake, and K. Yamaguchi, "An integrated ocr software for mathematical documents and its output with accessibility," in *Computers Helping People with Special Needs*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3118, pp. 648–655.
[5] R. Munavalli and R. Miner, "Mathfind: a math-aware search engine," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '06. New York, NY, USA: ACM, 2006, pp. 735–735. [Online]. Available: http://doi.acm.org/10.1145/1148170.1148348
[6] J. Mišutka, "Indexing mathematical content using full text search engine," in *WDS' 08 Proceedings of Contributed Papers: Part I - Mathematics and Computer Sciences*, 2008, pp. 240–244.
[7] M. Adeel, H. S. Cheung, and S. H. Khiyal, "Math GO! prototype of a content based mathematical formula search engine," *Journal of Theoretical and Applied Information Technology*, vol. 4, no. 10, pp. 1002–1012, 2006.
[8] K. Yokoi and A. Aizawa, "An approach to similarity search for mathematical expressions using MathML," in *Towards digital mathematics library (DML)*, 2009, pp. 27–35.
[9] M. Kohlhase and A. Franke, "Mbase: Representing knowledge and context for the integration of mathematical software systems," *Journal of Symbolic Computation*, vol. 32, no. 4, pp. 365–402, 2001.
[10] S. Jeschke, M. Wilke, M. Blanke, N. Natho, and O. Pfeiffer, "Information extraction from mathematical texts by means of natural language processing techniques," in *ACM Multimedia EMME Workshop*, 2007, pp. 109–114.
[11] T. Kudo, "Mecab: Yet another part-of-speech and morphological analyzer," http://mecab.sourceforge.net/.
[12] S. S. Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 807–814.
[13] N. Okazaki, "Classias: a collection of machine-learning algorithms for classification," 2009. [Online]. Available: http://www.chokkan.org/software/classias/

Semantic Aspect Retrieval for Encyclopedia

Chao Han, Yicheng Liu, Yu Hao, and Xiaoyan Zhu

***Abstract*—With the development of Web 2.0, more and more people contribute their knowledge to the Internet. Many general and domain-specific online encyclopedia resources become available, and they are valuable for many Natural Language Processing (NLP) applications, such as summarization and question-answering. We propose a novel encyclopedia-specific method to retrieve passages which are semantically related to a short query (usually comprises of only one word/phrase) from a given article in the encyclopedia. The method captures the expression word features and categorical word features in the surrounding snippets of the aspect words by setting up massive hybrid language models. These local models outperform the global models such as LSA and ESA in our task.**

***Index terms*—Aspect retrieval, online encyclopedia, semantic relatedness.**

I. Introduction

With the development of Web 2.0, more and more people contribute their knowledge to the Internet. Many general and domain-specific online encyclopedia resources become available, such as Wikipedia[1] and Baidu Baike[2] (the largest Chinese online encyclopedia website). They are well-organized by the categories and interrelations of their entries, meanwhile their content has relatively higher quality than general web pages. So these resources are valuable for many Natural Language Processing (NLP) applications, such as summarization and Question-Answering (QA).

In this paper, we only focus on a specific task: given a "entity-aspect" pair as query, we retrieve passages semantically related to the aspect word from the article corresponding to the entity in the encyclopedia. In the input, the "entity" must be a title of certain article in the encyclopedia; the "aspect" describes some attribute or sub-topic of the entity, and usually comprises of only one word or phrase. For example, for the entity-aspect pair "apple-nutrient", we retrieve the passages which describe apple's nutrient from the "apple" article in the encyclopedia.

The motivations of this task are as follows.

First, this task is an important approach for automatically answering complex natural language questions. A considerable proportion of questions can be converted into a simple description by an entity-aspect pair, as shown in Table I. We can answer this kind of questions directly by giving user the passages related to the aspect from the encyclopedia article corresponding to the entity.

Manuscript received November 1, 2010. Manuscript accepted for publication December 21, 2010.

The authors are with the Department of Computer Science and Technology, Tsinghua University, China (e-mail: hanc04@gmail.com).

[1]http://www.wikipedia.org.

Second, we choose passage as the unit we retrieve because passage retrieval is a very practical way to supply useful information to users in question-answering and information retrieval field. Usually, for a question answering system, returning users the exact answer is not the best choice [1], users would like to see some surrounding text to make sure that the answer is credible.

Third, because of the higher quality of online encyclopedia, the passages we retrieve can be used in some subtasks such as answer quality validation and so on.

TABLE I
CONVERSION FROM QUESTIONS TO ENTITY-ASPECT PAIRS

Question	Entity–Aspect Pair
What is the nutrient of apples?	apple - nutrient
How about the climate of China?	China - climate
What is a tiger like?	tiger - appearance
What causes diabetes mellitus?	diabetes mellitus - pathogenesis

Besides, to retrieve the passages from a given article of the encyclopedia is an interesting and useful task. Imagine this scenario: a mobile Internet user, who wants to know the nutrient of apple from Wikipedia, would scroll the small-size screen over and over to get what he wants if the search engine gives the whole article to her. It benefits users a lot if the system locates the screen at a more accurate position.

The difficulty lies on how to measure the semantic relatedness between the aspect word and the candidate passages. Simple method based on term vector space model does not work for two reasons: 1) The aspect word may not appear in the article. For example, in the "apple" article in Baidu Baike, the passage about "nutrient" is written as "*Apple contains a lot of pectin, which is a kind of soluble fiber, can make the content of cholesterol and bad cholesterol…*"[3], without using the word "nutrient" directly. 2) Even if the aspect word appears in some passage, the content of the passage may not be related to the aspect, either.

Considering those matters above, the method we adopt should satisfy at least three requirements as follows.

1) The method can handle arbitrary queries, and can measure semantic relatedness between short query and relatively long text.

2) The method should be unsupervised. Because the encyclopedia corpus is large, and it is hard to obtain large enough training set.

[2]http://baike.baidu.com.

[3]All texts from Baidu Baike are originally written in Chinese. We translate them into English in this paper.

3) The computing complexity of our method should be low because of the demand of fast response speed in online applications.

Besides, to certain extent, the method should have the ability of "rejection" when it is not quite confident for the answer.

In this paper, we propose a novel encyclopedia-specific method, which satisfies the requirements above, to retrieve passages for a given "entity-aspect" query. The method exploits features from category information and surrounding snippets. We compare the method with traditional semantic methods such as LSA [9] and ESA [10].

The remainder of the paper is organized as follows. In section 2 related work is discussed. In section 3 we present our approach. Section 4 is the experimental result. Finally, in section 5 we will conclude the paper.

II. Related Work

Nowadays, online encyclopedia websites assemble vast quantities of human knowledge. Consulting an online encyclopedia has become an importance approach for users to achieve the information they need.

Researchers have made great efforts to make it easy to utilize the online encyclopedia resource, especially Wikipedia. Ye et al. [2] proposed to summarize Wikipedia articles as definitions with various lengths to satisfy different user needs. Li et al. [3] proposed Facetedpedia, supplying users a faceted interface for navigating the result articles. The work of Hahn et al. [4] facilitates infobox data allowing users to query Wikipedia like a structured database through an attribute-value pairs extraction approach.

To the best of our knowledge, there is no previous work exactly on the task discussed in this paper. The key point of our task is to measure the semantic relatedness between the aspect word and candidate passages.

A lot of work has been done to quantify semantic relatedness of texts.

The work in [5] treats texts as bags of words and computes similarity in vector space. Lexical resources, such as WordNet, are used in [6] [7] [8].

Latent Semantic Analysis (LSA) [9] uses the Singular Value Decomposition (SVD) to analyze the statistical relationships among terms in a document collection. At first a matrix X with row vectors representing terms and column vectors representing documents, is constructed from the corpus. The cells of X represent the weights of terms in the corresponding documents. The weights are typically TF-IDF. Then SVD, which can be viewed as a form of principal components analysis, is applied to X, and the dimension is reduced by removing the smallest singular values. LSA measures the similarity of terms using the compressed matrix after dimension reduction, instead of the original matrix. The similarity of two terms is measured by the cosine similarity between their corresponding row vectors.

Explicit Semantic Analysis (ESA) [10] is a method based on concepts of Wikipedia or other corpus. ESA maps a text to a high-dimensional vector space with the value of each dimension representing the strength on some explicit concept, for example, an explicit concept can be a concept in Wikipedia. Then we can obtain the similarity between two texts by some measure such as the cosine value between the two corresponding vectors.

III. Our Approach

In our task, the main difficulty comes from the fact that the aspect query is too short – one word in most cases. So the first step of our approach seeks to expand the representation capacity of the aspect query.

TABLE II
SOME SURROUNDING SNIPPETS OF "NUTRIENT" IN ARTICLES OF CATEGORY FRUIT

1. Kiwi is rich in *vitamins* C, A, E in addition to *potassium*, *magnesium*, *cellulose*, but also contains other rare fruit **nutrient**s - *folic acid*, *carotene*, *calcium*, *progesterone*, *amino acids*, *natural inositol*. According to the analysis, every *100 grams* of Kiwi pulp will contain *100* to *300 milligrams* of *vitamin* C, 20 to 80 times higher than apple.
2. Lemon contains *citric acid*, *malic acid* and other *organic acids* and *hesperidin*, *naringin*, *Saint grass sub-glycosides* and other *glycosides*, also contains *vitamin* C, B1, B2 and *niacin*, *carbohydrates*, *calcium*, *phosphorus*, *iron* and other **nutrients**.
3. Citrus fruit is juicy and delicious, rich in *sugars*, *organic acids*, *minerals*, and *vitamins* and other **nutrients**. Its nutritional value is very high.

In Baidu Baike, each article is assigned to at least one category by its editors, and under each category, there are variant number of articles. For example, "pear" belongs to four categories: "fruit", "plant", "foodstuff" and "crops"; in category "fruit", there are about 1800 articles, such as "apple", "pear", "watermelon" and so on.

According to the characteristic of the way the encyclopedia articles are written and organized, we think that for an aspect word, the surrounding snippets of its occurrences contain the information we need to enrich the query. As shown in Table II, surrounding snippets of "nutrient" in articles of category "fruit" have some features in common.

We pick up two types of features.

The first type is **expression word feature**. In Table II, the underlined words, such as "contain", "pulp", "rich" and "high", are frequently used to help expressing the meaning of the content for the aspect.

The other type is **categorical word feature**. The italic words in Table II, such as "potassium", "iron" and "calcium", "citric acid" and "amino acids", are entities in the encyclopedia, and their category information is useful. For example, "potassium", "iron" and "calcium" may be used for the description of nutrients of different kinds of fruit, but they are all "chemical elements". The difference between these words and the expression words of the first type is that not

only the words themselves but also their categories are important: we use some chemical elements to describe the nutrients of certain fruit, whatever the chemical element is potassium or calcium.

A. Hybrid Language Model for Category-Aspect Pair

For certain category c and a potential aspect word w, we assemble them together as <*c*,*w*> and call it a category-aspect pair.

To utilize the surrounding snippets of the aspect word and capture the two types of features discussed above, we set up a hybrid language model HLM*c*,*w* for each category-aspect pair <*c*,*w*> as follows.

Step 1: Construct the surrounding snippets collection for <*c*,*w*>.

We index all Baidu Baike articles using Lucene[11]. For category-aspect pair <*c*,*w*>, search all occurrences of word *w* in the articles of category *c*, and extract all the surrounding snippets with length of 200 Chinese characters for each snippet.

In the surrounding snippets collection, as we can imagine, there exists a proportion of "outliers", which means the snippets contain the aspect word *w*, but the content of them are not related to the aspect; the occurrence of *w* here is "occasional". Thus we do a simple preprocess to reduce the influence of the outlier snippets: concatenate all snippets into a document *d*, then compute the cosine similarity under vector space model [5] between each snippet and *d*; then filter out at least 30% of snippets with smallest similarity values and save no more than 200 snippets.

Step 2: Build the language model $WLM_{c,w}$ for words information.

After obtaining the surrounding snippets collection through Step 1, we concatenate all snippets into a document *d*, with which infer a unigram language model $WLM_{c,w}$ [12].

For any text *p*, we have

$$P(p \mid WLM_{c,w}) = \prod_{i=1}^{n} P(t_i \mid WLM_{c,w}) \quad (1)$$

where t_i is the *i*th term (word) in the text *p*. And for each term *t*,

$$P(t \mid WLM_{c,w}) = \alpha \cdot \frac{tf(t,d)}{dl_d} + (1-\alpha) \cdot \frac{cf_t}{cs} \quad (2)$$

where α is a weighting parameter between 0 and 1, $tf(t, d)$ is the frequency of *t* occurs in *d*, dl_d is the document length of *d*, cf_t is the frequency *t* occurs in the entire collection, and *cs* is the total number of terms in the whole encyclopedia.

Step 3: Build the language model $CLM_{c,w}$ for categories information.

The difference between Step 2 and 3 is that the terms for language model $CLM_{c,w}$ are not words, but categories. For document *d* which is constructed by all snippets in Step 2, we do not use it to infer a language model directly. Instead, *d* which is a document consisting of words is mapped into a document *d'* consisting of categories by the procedure as follows:

Extract all the entries of Baidu Baike occurred in *d*, and add the categories of each entry into *d'*. For example, if "calcium" is found in *d*, we add its categories "metal", "chemical element", "nutriology" and "milk calcium" into *d'*.

After this mapping, we can obtain a category language model $CLM_{c,w}$ in similar way as Step 2.

For any text *p*, we first map *p* into *p'* in the same way document *d* is processed. Then we have

$$P(p' \mid CLM_{c,w}) = \prod_{i=1}^{n} P(c_i \mid CLM_{c,w}) \quad (3)$$

where c_i is the *i*th term (category) in the *p'*. And for each term *c*,

$$P(c \mid CLM_{c,w}) = \alpha \cdot \frac{tf(c,d')}{dl_{d'}} + (1-\alpha) \cdot \frac{cf_c}{cs} \quad (4)$$

where $tf(c, d')$ is the frequency of *c* occurs in *d'*, $dl_{d'}$ is the length of *d'*, cf_c is the number of articles belonging to category *c*, and *cs* is the total number of articles in the whole encyclopedia.

Step 4: Build the hybrid language model $HLM_{c,w}$.

The hybrid language model $HLM_{c,w}$ comprises two language model instances: $WLM_{c,w}$ and $CLM_{c,w}$, which are based on the surrounding snippets collection of *w* from the articles of category *c*.

For any text *p*,

$$P(p \mid HLM_{c,w}) = \lambda \cdot P(p \mid WLM_{c,w}) + (1-\lambda) \cdot P(p' \mid CLM_{c,w}) \quad (5)$$

where λ is the parameter to adjust the weights of two models.

B. Passage Ranking

Now get back to our task. Given the user query in the form of entity-aspect pair, such as "pear-nutrient", we have to compute the semantic relatedness score, denoted by *score*(*p*), between the aspect word *w* and each candidate passage *p* in the article. For one category-aspect pair <*c*,*w*>, we already know $P(p| HLM_{c,w})$. But usually there are more than one category for an entity, for example, "pear" belongs to four categories: "fruit", "plant", "foodstuff" and "crops". So the weight sum of $P(p| HLM_{c,w})$ for all categories should be used:

$$score(p) = P(p \mid w) = \sum_{i=1}^{k} P(p \mid HLM_{c_i,w}) \cdot P(HLM_{c_i,w} \mid w) \quad (6)$$

where c_i is the *i*th category of the entity, $i = 1, 2, \ldots, k$.

We estimate the conditional probability

$$P(HLM_{c_i,w} \mid w) = P(c_i \mid w) = \frac{P(w, c_i)}{P(w)} = \frac{P(c_i)P(w \mid c_i)}{\sum_{j=1}^{k} P(c_j)P(w \mid c_j)} \quad (7)$$

For each category c_i, we think they are equiprobable, i.e. $P(c_i) = 1/k$, $i = 1, 2, \ldots, k$. So we have

$$P(HLM_{c_i,w} \mid w) = \frac{P(w \mid c_i)}{\sum_{j=1}^{k} P(w \mid c_j)} \quad (8)$$

and

$$P(w \mid c_i) = \frac{df(w, c_i)}{cs(c_i)} \quad (9)$$

where $df(w,c_i)$ is the document frequency of w in all articles of category c_i, and $cs(c_i)$ is the total number of articles of category c_i.

The candidate passages are ranked by $score(p)$ in (6).

C. Model Database

For each category-aspect pair $<c,w>$, the construction of $HLM_{c,w}$ is a time-consuming procedure, because all the articles of category c is retrieved and searched. On average, about 4 to 10 seconds time is required for one query.

To make the algorithm available in online applications, we have to construct all the $HLM_{c,w}$ and store them into database in advance.

In Baidu Baike, there are totally 358,057 categories and more than 50,000 terms after removing stopwords and rare words. Thus the amount of category-aspect pairs is more than 1.79×10^{10}, which is a huge number we can't accept. So it is necessary to reduce the scale.

We only save the model for the category-aspect pair$<c,w>$ which satisfies the two conditions below:

1) The category c should contain at least 300 articles.

2) $P(w|c)$, as shown in (9), is larger than 0.3 and $df(w,c)$ is larger than 50.

There are 1660 categories which have at least 300 articles in Baidu Baike. The categories with small number of articles are almost rare and concerned by users by little chance or created by editors' mistakes.

By Condition 2, we reduce the scale of aspect words dramatically. The aspect words should reflect generality of entities under the same category. So we select aspect words by $P(w|c)$.

After the filtering procedure, the scale of the model database is reduced to less than one million, which is an acceptable value.

D. Rejection of Unreliable Answers

The methods based on LSA or ESA will always give an answer – the passage most related to the aspect word, whatever the entity-aspect query is, even if the query is meaningless such as "pear- pathogenesis", because they just compute and a result will come out finally for any situation. So it is very difficult to guarantee the quality of the result. Sometimes, a meaningless answer is much worse than no answer.

Our approach supplies a way to reject to give user answers with low confidence: if the $HLM_{c,w}$ models needed in (6) do not exist in the model database built in last section, the system can choose not to return any answer to users, because in this situation, w may not be a proper aspect word or our approach cannot handle it confidently.

IV. Experiments

A. Data Set

There is no open data set for the evaluation of our task, so we built the data set under the help from several volunteers.

First we collected more than 10,000 questions from Baidu Zhidao[4], which is a Chinese community question-answering website as Yahoo! Answers[5]. After a preliminary filtering by program, we picked out proper questions as those in Table I, and converted them into entity-aspect pair.

For each entity-aspect pair, we cut the corresponding article of Baidu Baike into passages. Each passage is a section or some continuous paragraphs with length no longer than 500 Chinese characters. The average number of passages of each article is 26.05.

Then the volunteers gave a label "related" or "unrelated" to each passage with respect to the aspect word. We totally labeled 411 queries. The average number of related passages for each query is 3.10.

We classify all queries into two types. For the queries of Type 1, the aspect word appears in the text of the article, while for the queries of Type 2, the aspect word does not appear in the article.

B. Analysis of the Results

We compared our approach with three methods: vector space model, LSA and ESA.

For all methods, we removed the stop words and the words with low frequency from text. The size of remaining word list is about fifth thousand. And for anyone of the methods, we didn't do any keyword expansion.

We trained the LSA and ESA model with the top 10,000 articles with the largest pagerank value in Baidu Baike. The dimension of LSA model is set to 200. And the weighting parameter λ in (5) for our approach is set to 0.2 empirically.

TABLE III
Result for all queries

Method	VSM	LSA	ESA	HLM
MAP@10	0.4901	0.5938	0.3836	**0.6835**
MRR@10	0.5466	0.6422	0.3946	**0.7518**
SUC@1	0.4185	0.4647	0.2728	**0.6302**
SUC@3	0.6204	0.7908	0.4541	**0.8491**
SUC@5	0.7129	0.8808	0.5661	**0.9270**

For evaluating the performance of each method, we use the classical metrics in information retrieval field: MAP@n, MRR@n and SUC@n.

MAP@n is the mean average precision for the first n results.

MRR@n is the mean reciprocal rank for the first n results.

[4] http://zhidao.baidu.com.
[5] http://answers.yahoo.com.

SUC@n is the mean success rate for the first n results. For each test case, it is one "success" if there is at least one result is labeled "related" in the first n results.

The results for all queries are shown in Table III. And the results for queries of Type 1 and 2 are shown in Table IV and Table V respectively. Our approach is noted as HLM.

From the results, we can see that our approach consistently outperforms all other methods including LSA. On SUC@1, which represents the success rate at the first result, our approach performs significantly higher than other methods for about 20 percent.

Comparing the results in Table IV and Table V, the performance of all methods drops to certain extent. VSM turns to a random ranking in Table V, because without the aspect word appearing in the text, VSM can't distinguish any passages.

Even for queries of Type 1, HLM is better than VSM. This is because even if the aspect word appears in some passage, the content of the passage may not be related to the aspect, either. The appearance can be an outlier.

TABLE IV
RESULT FOR QUERIES OF TYPE 1

Method	VSM	LSA	ESA	HLM
MAP@10	0.6912	0.6949	0.4176	**0.7353**
MRR@10	0.7896	0.7505	0.4252	**0.8297**
SUC@1	0.6862	0.5904	0.3138	**0.7340**
SUC@3	0.8723	0.9043	0.4947	**0.9149**
SUC@5	0.9255	0.9681	0.5826	**0.9734**

TABLE V
RESULT FOR QUERIES OF TYPE 2

Method	VSM	LSA	ESA	HLM
MAP@10	0.3205	0.5086	0.3550	**0.6398**
MRR@10	0.3418	0.5509	0.3688	**0.6862**
SUC@1	0.1928	0.3587	0.2383	**0.5426**
SUC@3	0.4081	0.6951	0.4198	**0.7937**
SUC@5	0.5336	0.8072	0.5522	**0.8879**

It is worth noticing that ESA performs badly in this task. We think the reason lies in the fact that ESA uses the articles themselves as concepts directly. It is easy to understand that two different aspect words w_1 and w_2 for category c may have a lot of co-occurrence in the articles under category c. So ESA cannot distinguish w_1 and w_2 easily.

V. CONCLUSION

We propose a novel encyclopedia-specific method to retrieve passages which are semantically related to an aspect query from a given article in the encyclopedia. The method captures the expression word features and categorical word features in the surrounding snippets of the aspect words by setting up massive hybrid language models. These local models outperform the global models such as LSA and ESA. By store these models into database in advance, we make a trade-off between time cost and space cost so as to make the method usable for online situation. In addition, our approach has the ability to reject to give user answers with low confidence.

REFERENCES

[1] J. Lin, D. Quan, V. Sinha, K. Bak-shi, D. Huynh, B. Katz, and D. R. Karger, "The role of context in question answering systems," in *Proceedings of the 2003 Conference on Human Factors in Computing Systems*, 2003.

[2] S. Ye, T. Chua and J. Lu, "Summarizing Definition from Wikipedia," in *Proceedings of the 47thAnnual Meeting of the ACL. Singapore*, 2009.

[3] C. Li, N. Yan, S. B. Roy, L. Lisham and G. Das, "Facetedpedia: Dynamic Generation of Query Dependent Faceted Interfaces for Wikipedia," in *Proceedings of International World Wide Web Conference, Raleigh, North Carolina, USA*, 2010.

[4] R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Brgle, H. Dwiger, and U. Scheel, "Faceted Wikipedia Search," in *13th International Conference on Business Information Systems (BIS)*, 2010.

[5] R. B. Yates and B. R. Neto, *Modern Information Retrieval*, Addison Wesley, New York, NY. 1999.

[6] C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA. 1998.

[7] A. Budanitsky and G. Hirst, "Evaluating Wordnet-based Measures of Lexical Semantic Relatedness," *Computational Linguistics*, 2006, pp. 13-47.

[8] P. Roget, *Roget's Thesaurus of English Wordsand Phrases*, Longman Group Ltd., 1852.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harsh-man, "Indexing by Latent Semantic Analysis," *Journal of the American Society For Information Science*, 1990, pp. 391-407.

[10] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis," in *Proceedings of IJCAI*, 2007, pp. 1606-1611.

[11] E. Hatcher and O. Gospodnetic, *Lucene in action*, Manning Publications, 2005.

[12] J. M. Ponte, and W. B. Croft, "A Language Modeling Approach to Information Retrieval," in *Proceedings of the 21st Intl. ACM SIGIR Conf.*, 1998, pp. 275-281.

Are my Children Old Enough to Read these Books? Age Suitability Analysis

Franz Wanner, Johannes Fuchs, Daniela Oelke, and Daniel A. Keim

***Abstract*—In general, books are not appropriate for all ages, so the aim of this work was to find an effective method of representing the age suitability of textual documents, making use of automatic analysis and visualization. Interviews with experts identified possible aspects of a text (such as 'is it hard to read?') and a set of features were devised (such as linguistic complexity, story complexity, genre) which combine to characterize these age related aspects. In order to measure these properties, we map a set of text features onto each one. An evaluation of the measures, using Amazon Mechanical Turk, showed promising results. Finally, the set features are visualized in our age-suitability tool, which gives the user the possibility to explore the results, supporting transparency and traceability as well as the opportunity to deal with the limitations of automatic methods and computability issues.**

***Index Terms*—Information interfaces and presentation, information search and retrieval.**

I. Introduction

TWITTER messages, blog posts, customer reviews, and other user-generated content in the internet provide a wealth of information for companies and potential customers to learn about the strengths and weaknesses of different products. Studies have shown that about 81% of the Internet users in the U.S. have done online research on a product at least once [1]. In the last years, many text analysis approaches were developed that support the user in mining these resources. Automatic algorithms for opinion and sentiment detection permit to process a set of customer reviews automatically and present a summary of the product's most liked or disliked features.

This approach works well for many types of products. However, there are purchase decisions that are not adequately supported by the available methods. For example, before buying a book many potential readers would like to see if the writing style suits their taste. Some online stores meet this need by offering a "Look Inside" functionality that allows you to read some pages of the book. But this often is not enough to determine what age a book is suitable for. To assess this more than just the writing style needs to be taken into account.

For many books, the retail market and sometimes also the publishers provide a recommendation for the reader's age. However, often this is arguable. For example, the whole series of "Harry Potter" is recommended as being suitable for readers at the age of 9 to 12. Critics remarked that there is clear increase in violence and blood-curdling fragments in the later books of the series. Furthermore, the length of the book changed from 300 pages in the first volume to more than 780 in the final book of the series. It was therefore encouraged to rethink whether the books should really be all recommended for the same age range. Our interviews in german book stores confirmed this impression: at least some retailers shared this subjective view about the book.

Asked what aspects should be taken into account when determining the age group that a book is suitable for, the interviewed retailers suggested to take a look at the following parameters: (a) The difficulty of the writing style, (b) the complexity of the story, (c) the topics that are covered, (d) the emotions that are aroused, and finally (e) the ratio between pictures and textual content as well as other physical aspects such as the font size that is used.

In this paper, we present an approach that computationally assesses these five aspects. Rating books with an automatic algorithm comes with the advantage that it is independent of economic interests and individual opinions and positions. By measuring the different aspects separately and subsequently visualizing the result, it becomes possible to weight the different influences as desired. This permits to take individual preferences and special needs of the reader into account.

The paper is structured as follows: After a discussion of related work in section II, we introduce the different features for measuring age suitability in section III. With the help of the Amazon Mechanical Turk [2], a ground-truth data set was established that is then used in section IV to evaluate the features. Finally, a multi-view dashboard visualization is provided that allows the user to explore the detailed information that was extracted about the book (section V). Section VI concludes the paper.

II. Related Work

A. Related Work for Features Approximating Age Suitability

Subjectivity analysis is the recognition of opinion-oriented language in order to distinguish it from objective language. Sub-areas of subjectivity analysis are opinion or sentiment analysis. Many approaches and definitions can be found in [3]. However, the detection of emotion is slightly different. Important here is the determination of the expressed emotion. In [4] and [5] this was done for web news. The work in the

Manuscript received October 27, 2010. Manuscript accepted for publication January 28, 2011.

The authors are with the University of Konstanz, 78457 Konstanz, Germany (e-mail: wanner@dbvis.inf.uni-konstanz.de, Johannes.Fuchs@uni-konstanz.de, oelke@dbvis.inf.uni-konstanz.de, keim@dbvis.inf.uni-konstanz.de).

area of topic detection is tremendous and the focus lies on methods to detect and track events automatically. However, our goal is to get the specific topic of a book. Nallapati [6] compared the content of news articles by means of four categories. When the categories overlap sufficiently, then the compared documents build a topic. Another approaches are more appropriate for our needs determing topics in advance. The text classification algorithms of Green [7], Scott [8] or Hotho et al. [9] use WordNet, a lexical database. The advantage of such an approach is to provide semantical knowledge to the classification algorithm. Further methods and techniques can be found in the book of James Allan [10]. Text properties can be special in the sense that they do not measure a property that is in the text, but rather an "effect" that is caused by the text [11]. The story complexity can be seen as an effect, caused by many different characters and a fragmented story. Beside the already introduced readability of Oelke et al.,there are different algorithms to determine the readability of textual documents. Popular ones amongst others are the Gunning Fog [12] or the Flesch-Kincaid Readability Test [13]. It is common to all these measures that they base on statistical characteristics of the analyzed text. Additionally, we measure the vocabulary richness. This has been mainly used in the area of authorship attribution, for example [14] or [15].

B. Visual Approaches for Document Analysis

Full automatic algorithms hit their limit when human knowledge is required and in order to understand a document, knowledge of the world and human interpretation is needed [16]. This is the point where *Visual Analytics* can help. The aim of Visual Analytics is to make the way of processing data and information transparent for an analytic discourse [17]. Thereby, Visual Analytics helps the user gaining insight in the used algorithms and methods. In detail, the collaboration between the human and the computer is most important in our application in the analysis step, where the human's abilities to interpret and assess the results are in demand. Based on that, several work has been done in recent years. Combined with visualizations Oelke and Keim [18] showed in 2007 a new method for Visual Literary Analysis, which is called *Literature Fingerprinting*. The fingerprints are pixel-based visualizations, encoded with colour to show the text features. Tag clouds or word clouds have become more and more in use through the development and applications on the internet. These frugal text visualizations map the word frequency on font size [19]. The success of tag clouds in recent years is due to the fact, that users were allowed to create word clouds with their own content. One of the most famous single-purpose tool for example is wordle [20]. A more general visualization sharing site for example is Many Eyes [21]. It was generally created for explorative data analysis. Wordle is also able to support non-experts to visualize and arrange personally meaningful information [22]. A possibility to enrich word clouds with more information showed Wanner et al. [23]. POSvis [24] is an example for Literature Analysis using a tag cloud amongst others. The authors tried to analyze the book *The Making of Americans*. According to a specialist, the postmodern writing is very hard to read. The various visualizations (bar chart, text snippets) are arranged around a part-of-speech word cloud on a dashboard. Additionally, the software allows the user to explore and analyze the document. We are also use such visualization techniques and give the user the possibility to explore and detect interesting parts of the book.

III. Features To Measure Age Suitability

As mentioned in the previous section, we could identify five different aspects of age suitability in our interviews with booksellers. For each of these properties we separately define a measure to approximate them computationally.

A. Linguistic Complexity Feature

Linguistic complexity can be measured in terms of the vocabulary that is used or with respect to the ease of reading. Measures of vocabulary richness are mainly based on the evaluation of the number of different types (unique vocabulary items) and the overall number of tokens (any occurrence of a word type, i.e. the text length). In this work, we make use of the *Simpson's Index (D)* [14] that calculates the probability that two arbitrarily chosen words belong to the same type.

$$D = \frac{\sum_{r=1}^{\infty} r(r-1)V_r}{N(N-1)}$$

In the formula, N denotes the number of tokens (i.e. the text length) and V_r the number of vocabulary items that occur exactly r times.

To assess the readability of the text, the *Automatic Readability Index*[25], a popular readability measure, is used. It consists of two parts: (a) an estimation of the difficulty of the words that are used (assuming that longer words are more difficult to use) and (b) the average sentence length as an indicator for the difficulty to process the sentence.

$$ARI = 4.71 \cdot (\frac{\#characters}{\#words}) + 0.5 \cdot (\frac{\#words}{\#sentences}) - 21.43^{1}$$

The measure is normalized in a way that the resulting values range between 1 and 12, reflecting the US grade level that is needed to understand the text.

B. Story Complexity Feature

Measuring the complexity of a text on a statistical and syntactic level is reasonable and important, however, there are more factors that contribute to complexity. Next, we are going to look at the discourse level of the text by assessing the complexity of the story line. Measuring text properties on a higher linguistic level than the statistical level is challenging. Usually, there is no way to measure these aspects directly.

[1] # denotes "number of"

We therefore have to identify aspects that contribute to the specific property and approximate them with features that are computationally accessible.

Since we are assessing the complexity of novels, an analysis of the characters of the novel suggests itself. Are there one or several main protagonists or do the most important characters change from chapter to chapter? And how many characters exist in total? Are there groups of characters that are always mentioned together? Our assumption is that a story becomes more complex if many characters exist and there is no main protagonist that the reader can follow through the story. Furthermore, it has to be assumed that a frequent change in the relations of the characters adds complexity to the story compared to a situation in which distinct groups exist that always occur together.

However, so far this is just an assumption and we do not know how much the different aspects contribute to the story complexity. Section V illustrates how visual analysis can help to overcome this gap between the statistical level that can be accessed computationally and the semantics of a text that we would like to measure instead.

A central requirement of this measure is that we are able to extract the characters of a novel automatically. Our algorithm consists of three steps: First, the candidate extraction, second, a filtering step to extract only those characters who play an active role and finally, the classification of a name as first name, middle name, or last name. The resulting list can then be used to identify all occurrences of active protagonists within the novel.

To get a candidate set of names, we first used a common named entity recognition algorithm like the Stanford NER [26] to extract all persons in the text. In the next step the received characters which are not at least once followed or preceded by a communication verb are dismissed. Communication verbs are verbs such as “say”, “tell”, or “ask” that describe a communicative action by a person [27]. Using these terms, we can dismiss characters that do not play an active role in the plot (and therefore also do not contribute to the story complexity).

If an active protagonist was directly followed or preceded by an other person, the whole noun phrase was extracted as a character name in step 1 of the process. In this final step of the algorithm, we now try to identify the full names of the protagonists and filter out incomplete duplicates in our list. Following again [28] this is done with the help of a few simple rules. If a noun phrase consists of two terms, we mark the first one as the *first name* and the second one as the *last name* of the character. In case of three nouns, the middle one is classified as *middle name*. If an extracted term consists of only one token, we do not know whether this is the first name or the last name of the character. Often it is possible to resolve this ambiguity in the course of the process if at some other place the full name of the person is mentioned. If no such resolution is possible (e.g. because the full name never occurs or the decision cannot be made unambiguously, because there are multiple characters with the same first or last name) the name is classified as *unique* and treated as a separate name.

C. Topic Feature

To learn about the topic of the book, we analyze also its semantical content. For each topic that we would like to analyze, we need a word list with typical terms. Thereby, we restrict ourselves to topics that have an impact when analyzing a book with respect to age suitability. We chose to take the following topics into account: war, crime, sex, horror, fantasy, and science fiction. For each one we compiled an initial term list with indicative nouns and verbs. To calculate a score for each topic we extend every word in the text by a) adding synonyms and b) adding hypernyms. Both can be done with the help of WordNet [29], a lexical semantic network that is based on synsets of words. When adding hypernyms stopping at the right hierarchy level is critical in order to avoid over-generalizations (see [8] for a more detailed discussion). In the next step the extended word list is compared with the respected topic list counting every occurrence and normalize this with the overall number of words in the text. To account for the fact that some terms are more discriminative than others, we make use of the Brown Corpus B [30] which contains the most frequent 2000 English terms. Terms that can be found in this list are down-rated by a user-specified factor α (with $0 < \alpha < 1$) when counting the number of topic-related terms in a text unit. This last step is important because many words in the general linguistic usage are associated with the topic war because of adding hypernyms from WordNet.

D. Emotion Feature

The age that a book is suitable for is also affected by the emotions that are aroused by its content. Measuring this aspect directly is not possible. However, looking at the meaning of the words that are used, we can draw conclusions about emotional aspects. In our measure we therefore make use of a list of emotional words that were collected and rated during a psychological experiment at the University of Reading [31]. In the list, four categories of terms exist: Happiness, sadness, anxiety, and anger. Each category exists of 30 representing words and enriched with associations. The negative associated terms were dismissed because they would falsify the result as in the example: love associated with hate or happy associated with sad. Like with the topic feature, we calculate a score for each category by counting how many of its terms are mentioned in the text. These values are then normalized with the total number of terms in the document to permit a comparison of values between different books.

E. Book Dimension Feature

Finally, we take a look at the dimensions of the book. Parameters such as the font size, the ratio between pictures and textual content, and the number of pages of the book can

provide valuable information about the age group that a book was designed for. The necessary data can be retrieved from online databases.

IV. Evaluation

The evaluation of the different features is done separately. The following two sub-sections handle the Story Complexity and the Topic Detection. The Readability will not be evaluated because no new algorithm was implemented.

The fact that our data consist of whole books make it impossible to get objective ground-truth data. Publisher suggest a minimum age for every book but are perhaps influenced by economic reasons. That is why it was necessary to generate our own ground-truth data. Therefore we used a so called Human Intelligence Task (HIT) with the Amazon Mechanical Turk Service. This service provides a crowd-sourcing marketplace to execute different types of tasks by ordinary people. A single HIT is an online job which can be executed by every Amazon Mechanical Turk member fulfilling the requirements. Our HIT consists of a questionnaire with 14 questions about 15 different books. At least the questions to one book must be answered to receive a small award. Every answer was checked for trustworthiness examining an implemented time stamp and the correlation between two test questions. About 300 questionnaires were answered trustworthy and provide our ground-truth data. Only six of the 15 books were answered often enough to be analysed to guarantee the methodological correctness.

A. Evaluation of Story Complexity

For the evaluation we took the book *Harry Potter and the Philosopher's Stone* with a total of 179 Characters. Following you can see our results:

TABLE I
Results of evaluation

	Relevant	Non Relevant
Retrieved	69	29
Not Retrieved	47	34
Total	116	63

The precision of the algorithm is 0.704 and the recall 0.595. When we took a look in our results we recognized, that the NER process is not consistent over the book. So "Hagrid", a character of the Harry Potter series, is tagged as *person* and elsewhere in the book as an *organization*. If the right tagged noun is never at least once followed or preceded by a communication verb, but the wrong one does so our result gets worse. A solution could be implementing a threshold, e.g. as a hypothesis "Hagrid" is detected 75 percent as a *person* and 25 percent as *organization* then we could assume that "Hagrid" is a person. Although, that could lead to problems (e.g. "Washington") an improvement could be achieved. We would like to try that in the future.

B. Evaluation of Topic Detection

Our implemented algorithm to compute the possibility that a certain book belongs to a specific topic will be evaluated using the answers of the online questionnaire as our ground-truth data. The participants had to choose whether the book is about one or more of the six predefined topics or not. To compare our algorithm with the user opinion the results were normalized between 0 and 1. Additionally the significance value used in our algorithm is examined. Each book is therefore analyzed twice once with the significance value and another time without. The following figure illustrates the evaluation with four different books (Fig. 1).

The bar charts illustrate that the user tendency is much more similar to the algorithm with the significance value than without. However there are exceptions like the book *1984* (bottom left) where both results are misleading. The main part to improve the algorithm are the predefined hardcoded lists of representing words for each topic. With the lists being more complete and correct the whole algorithm performs better.

V. Visual Book Analysis

With the measures that were defined in section III we are able to approximate the different aspects of age suitability computationally. However, it is unclear how much each feature contributes to the overall rating. Furthermore, for some features we do not have a single score but a whole bunch of information that requires interpretation. We therefore decided to make use of visual analysis techniques in the next step of the analysis process. This comes with the following advantages:

- The human visual system is very powerful allowing the user to grasp a large amount of data at an instance as long as it is meaningfully displayed. [32] Visualization therefore is an ideal means of integrating the user into the process.
- Thus, using visualization allows us to provide the detailed information of our measures to the user without causing too much cognitive load.
- It is known that humans are very proficient in detecting visual patterns, a capability that is highly desirable in this case because of the complex measures that are used. With this, the interpretation of the data that is needed to overcome the semantic gap can be left to the human analyst.
- At the same time this comes with the advantage that the human analyst does not need to trust a "black box" but is able to comprehend the decision of the algorithm. This is especially important for features that may be weighted differently depending on the personality of the reader.

In the following, we are going to introduce our visual analysis tool. As the emotion detection and the analysis of the story complexity are the two features that profit most from the visual analysis, their visualizations are presented in detail in sections V-A and V-B. This is followed by a presentation of the full application.

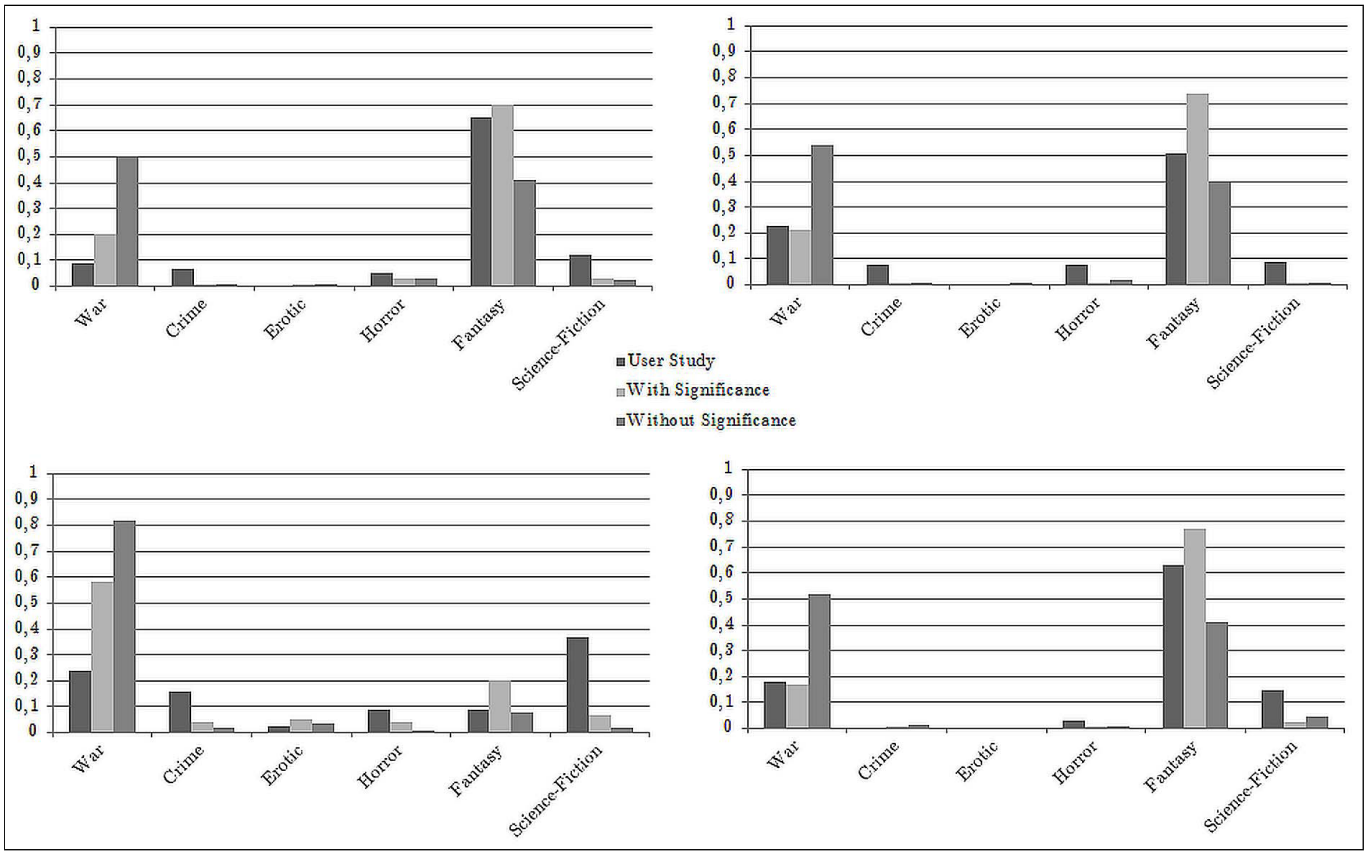


Fig. 1. Comparison of the Topic Detection of the books *Harry Potter and the Philosopher's Stone* (top left), *Harry Potter and the Deathly Hallows* (top right), *1984* (bottom left) and *The Hobbit or There and Back Again* (bottom right).

A. *Visualization of Story Complexity*

Our Story Complexity feature counts and detects the characters in the text and tags their position. This allows us to track the different characters across the text and to analyze who is interacting with whom. What we are especially interested in is whether there is a consistent story line (according to the characters) or if many different persons show up in changing combinations.

To arrange the different characters in a clear way we changed Oelkes Summary Report visualization [33] to fit our task. The following graphic illustrates the analysis with the character feature for the book *Harry Potter and the Philosopher's Stone*.

Each row represents one character and each column handles one text unit (e.g. a chapter). The seven most frequent characters are shown in the top color-coded lines. This is followed by a line that summarizes all the rest of the characters. The size of the inner rectangles in this grey line hints at the number of persons that are represented by this symbol. The user can manually change the number of single lines representing one character. The saturation of a rectangle is determined by the number of times that the name is mentioned in the corresponding text unit. If the character does not show up in one of the sections, the corresponding rectangle remains empty.

With this encoding, the user is enabled to compare the occurrences of different persons across the book. For example in Fig. 2 the orange rectangles are filled in nearly every section illustrating that this character is mentioned in every section. Interestingly, the characters *Dumbled* and *Dudley* next to never appear in the same section. We can conclude from this that they did not interact with each other in the story.

The comparison in Fig. 3 illustrates the differences between a more complex text and an easier one. In the upper graphic there is one character (depicted in orange) that is acting over the whole text. At some point people interact with him and accompany him through some parts of the story. In the graphic beneath no main protagonist can be discerned. Only rarely two of the seven most frequent characters are mentioned in the same section. The long sparsely colored passages show that the seven most frequent characters do not provide enough details for an analysis of this novel. Therefore it is necessary to reveal a few more characters to get an insight into the more divided protagonists.

This arrangement suggests that there is more than just one story line in the novel which very likely accounts for a higher complexity.

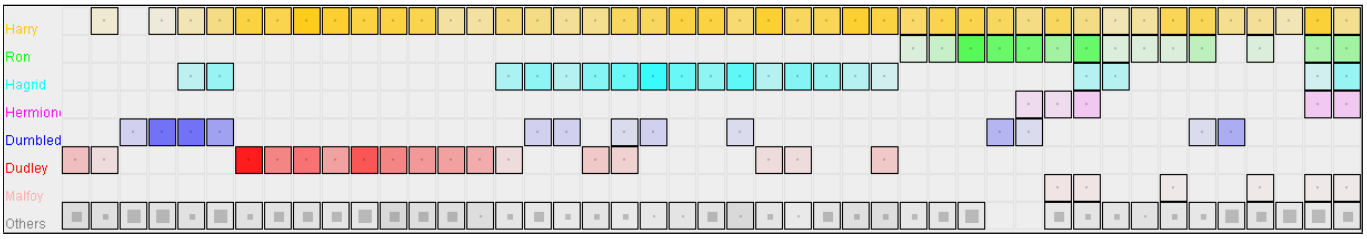


Fig. 2. Story Complexity Visualization of the book *Harry Potter and the Philosopher's Stone*

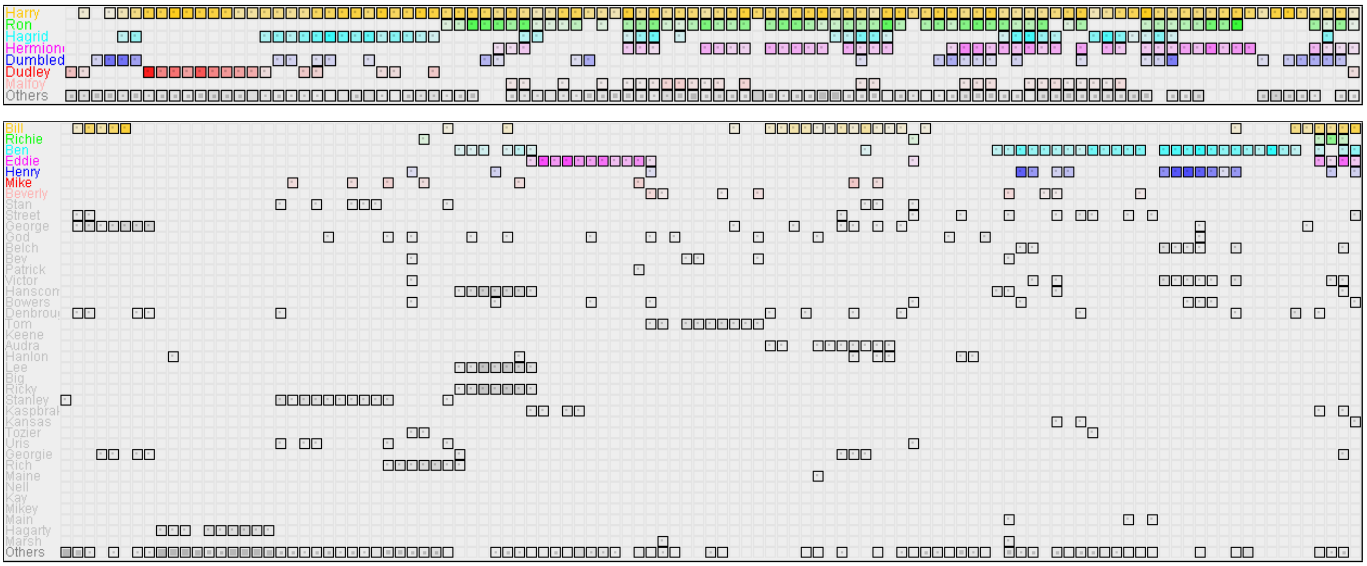


Fig. 3. Comparison of the Story Complexity Visualization of the books *Harry Potter and the Philosopher's Stone* (top) and *It* (bottom).

B. Visualization of Emotions

The four different emotions happiness, sadness, anger and anxiety are visualized in a bar chart diagram. The height of each bar represents the number of detected emotion words for the specific category.

Especially with the emotion feature we are facing the challenge that we need to overcome a gap between what we measure and what we would like to approximate on a semantic level. Remember that we are interested in the *aroused* emotions but can only work with a measure that is based on word associations that are related to emotional states. Thus, an inspection and interpretation of the result by a human expert is critical. We therefore do not only visualize the overall emotion scores, but again calculate separate values for each text unit as for the story complexity. This also gives us the chance to analyze the development of the emotions across the text.

Fig. 4 shows the course of the emotion feature for the book *A Long Way Down*. This detailed view reveals much information about the story. While happiness is the most dominant emotion in most of the book, there is a passage in the middle in which it almost completely disappears. Furthermore, there are several text units in which sadness and happiness (red and yellow bars) occur with a similar strength suggesting that this might be an emotionally demanding part of the book in which the two contrasting emotions are close together. However, at the end of the story the happiness value is clearly dominating which hints at a happy end. Emotion words related to anger are nearly not present at all whereas anxiety is present at a certain level almost all over the book. To investigate a single bar chart in detail, it is possible to display a word cloud of the underlying emotion words (see figure 4).

C. Visual Agesuitability Tool

The final Visual Agesuitability Tool combines the visual representations of the five features in one multi-view dashboard display (see figure 5).

In the upper left corner, a summary of the detected emotions is presented in a bar chart diagram. Users can interactively drill-down to the detailed representation that is presented in section V-B. Similarly, the character panel at the bottom shows an overview representation of the active characters which can be zoomed in to get the in-depth information that is provided by the summary report visualizations that are depicted in figures 2 and 3. Numeric information such as the readability scores, the vocabulary richness, the number of pages, or the number of words per page are shown in the upper middle of the panel. Additionally, color is used to visually encode the numbers and support the user in assessing how these values

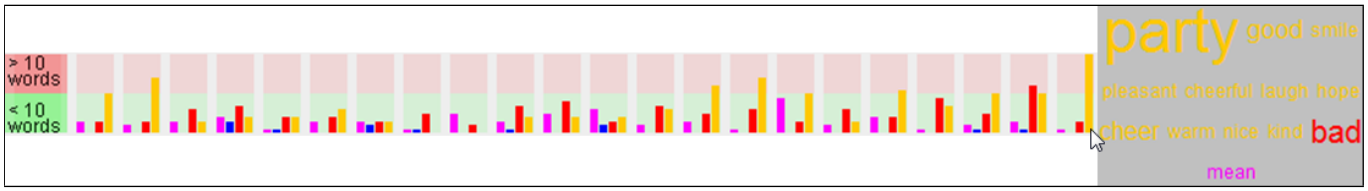


Fig. 4. Emotion Visualization of the book *A Long Way Down*. Anxiety = magenta, anger = blue, sadness = red, happiness = yellow

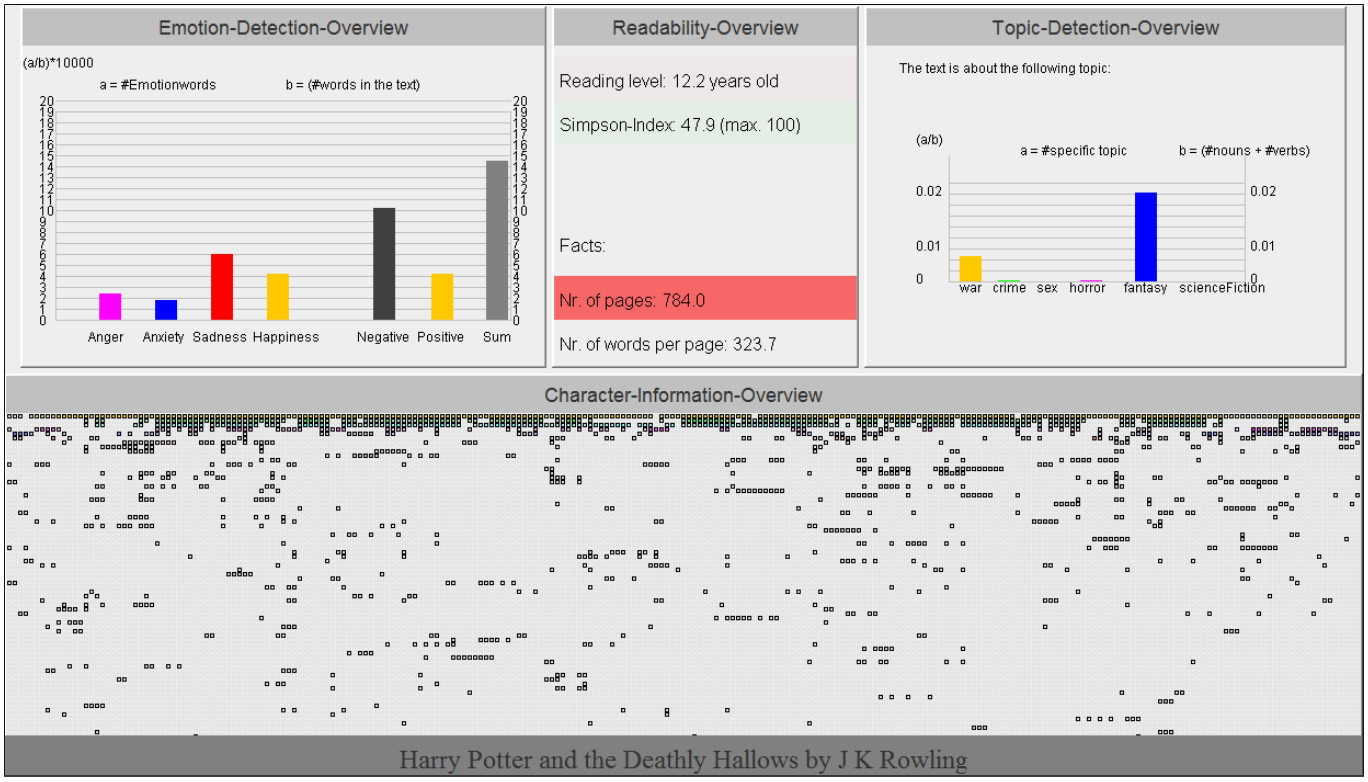


Fig. 5. Age Suitability Visualization of the book *Harry Potter and the Deathly Hallows*. In the upper left corner: distribution of emotions, in the middle: readability overview, upper right side: the topic(s) of the book. The lower part shows the occurence of the various characters in the book. Each line reflects a character. The topmost line is Harry Potter, the main character in the book. He occures in almost each text section.

range in comparison to other novels. For that, a color scale from red to green is used with red hinting at difficulties and green signalling that the text is comparably easy with respect to this feature.

Finally, the detected topics are visualized in a bar chart diagram. Thereby, the height of the bars depicts the influence of each topic as measured with the topic feature. Advanced interaction techniques such as brushing-and-linking enable the user to compare ratings across the different sections e.g. by marking a section in one of the visualizations that is then automatically highlighted in all the other visualizations.

VI. Conclusions

In this paper, we presented an approach for assessing the age suitability of a novel. We proposed to measure the different aspects of age suitability separately to provide a transparent, expressive feature that allows a detailed analysis of a book with respect to this higher-level text property. While for some of the sub-features such as the linguistic complexity or the analysis of the book dimensions standard measures could be used or a straight-forward approach exists, other features required some deeper consideration. For topic detection the use of a significance value has proven beneficial for the task.

The analysis of the novels with respect to story complexity and the emotions that are aroused came with a special challenge because these features cannot be measured directly.

We addressed this problem by providing expressive visualizations that allow the user to analyse the novels in detail and permit to defer the relevant information by interpreting the result of the automatic algorithm.

Furthermore, the proposed multi-view dashboard visualization shows all features at a glance, thereby offering the prospective reader or analyst a comprehensive overview with respect to the different aspects of age suitability.

REFERENCES

[1] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] J. Kadhim and V. Crittenden, "Amazon Mechanical Turk," retrieved from Citeseer.

[3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, pp. 1–135, January 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1454711.1454712

[4] J. Zhang, Y. Kawai, T. Kumamoto, and K. Tanaka, "A novel visualization method for distinction of web news sentiment," in *Web Information Systems Engineering - WISE 2009*, ser. Lecture Notes in Computer Science, G. Vossen, D. Long, and J. Yu, Eds. Springer Berlin / Heidelberg, 2009, vol. 5802, pp. 181–194.

[5] M. L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner, "User-directed sentiment analysis: visualizing the affective content of documents," in *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, ser. SST '06, 2006, pp. 23–30.

[6] R. Nallapati, "Semantic language models for topic detection and tracking," in *NAACLstudent '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003, pp. 1–6.

[7] S. Green, "Building hypertext links in newspaper articles using semantic similarity," in *Third Workshop on Applications of Natural Language to Information Systems (NLDB'97)*, 1997, pp. 178–190.

[8] S. Scott and S. Matwin, "Text classification using WordNet hypernyms," in *Use of WordNet in Natural Language Processing Systems: Proceedings of the Conference*, 1998, pp. 38–44.

[9] A. Hotho, S. Staab, and G. Stumme, "Wordnet improves text document clustering," in *Proc. of the SIGIR 2003 Semantic Web Workshop*. Citeseer, 2003, pp. 541–544.

[10] J. Allan, Ed., *Topic detection and tracking: event-based information organization*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[11] D. Oelke, D. Spretke, A. Stoffel, and D. A. Keim, "Visual readability analysis: How to make your writings easier to read," in *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST '10)*, 2010.

[12] R. Gunning, *The technique of clear writing*. McGraw-Hill, 1952.

[13] J. P. Kincaid, R. P. Fishburn, R. L. Rogers, and B. S. Chissom, "Derivation of New Readability Formulas for Navy Enlisted Personnel," Naval Air Station Memphis, Research Branch Report 8-75, 1975.

[14] D. I. Holmes, "Authorship Attribution," *Computers and the Humanities*, vol. 28, pp. 87–106, 1994.

[15] D. Hoover, "Another perspective on vocabulary richness," *Computers and the Humanities*, vol. 37, pp. 151–178, 2003.

[16] D. Oelke, "Visual document analysis: Towards a semantic analysis of large document collections," Ph.D. dissertation, University of Konstanz, 2010.

[17] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering the information age—solving problems with visual analytics*. Eurographics Association, 2010.

[18] D. A. Keim and D. Oelke, "Literature fingerprinting: A new method for visual literary analysis," in *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology (VAST '07)*. IEEE Computer Society, 2007, pp. 115–122.

[19] F. B. Viégas and M. Wattenberg, "Timelines tag clouds and the case for vernacular visualization," *interactions*, vol. 15, no. 4, pp. 49–52, 2008.

[20] "wordle, http://www.wordle.net/, october 31st, 2010."

[21] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, "Manyeyes: a site for visualization at internet scale," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1121–1128, 2007.

[22] F. B. Viégas, M. Wattenberg, and J. Feinberg, "Participatory visualization with wordle." *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1137–1144, 2009.

[23] F. Wanner, M. Schaefer, F. Leitner-Fischer, F. Zintgraf, M. Atkinson, and D. A. Keim, "Dynevi - dynamic news entity visualization," in *Proceedings of the International Symposium on Visual Analytics Science and Technology (EuroVAST 2010)*, J. Kohlhammer and D. A. Keim, Eds., Jun. 2010, pp. 69–74.

[24] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar, "What's Being Said Near "Martha"? Exploring Name Entities in Literary Text Collections," in *IEEE Symposium on Visual Analytics Science and Technology (IEEE VAST)*, Oct. 2009, pp. 107–114. [Online]. Available: http://liris.cnrs.fr/publis/?id=4360

[25] R. Senter and E. Smith, "Automated Readability Index," 1997, technical Report.

[26] D. Klein, J. Smarr, H. Nguyen, and C. Manning, "Named entity recognition with character-level models," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, 2003, pp. 180–183.

[27] Ubiquitous Knowledge Processing (UKP) Lab, TU Darmstadt, English communication verbs, www.ukp.tu-darmstadt.de/fileadmin/user_upload/Group_UKP/data/english_communication_verbs.txt.

[28] H. L. Chieu and H. T. Ng, "Named entity recognition: a maximum entropy approach using global information," in *Proceedings of the 19th international conference on Computational linguistics*, 2002, pp. 1–7.

[29] C. Fellbaum, *WordNet: An electronic lexical database*. MIT Press, 1998.

[30] "Frequency list from the brown corpus, www.edict.com.hk/lexiconindex/frequencylists/words2000.htm."

[31] C. John, "Emotionality ratings and free-association norms of 240 emotional and non-emotional words," *Cognition & Emotion*, vol. 2, no. 1, pp. 49–70, 1988.

[32] C. Ware, *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2004.

[33] D. Oelke, M. Hao, C. Rohrdantz, D. Keim, U. Dayal, L. Haug, and H. Janetzko, "Visual opinion analysis of customer feedback data," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, 2009, pp. 187–194.

Linguistically Motivated Negation Processing: an Application for the Detection of Risk Indicators in Unstructured Discharge Summaries

Caroline Hagege

***Abstract*—The paper proposes a linguistically motivated approach to deal with negation in the context of information extraction. This approach is used in a practical application: the automatic detection of cases of hospital acquired infections (HAI) by processing unstructured medical discharge summaries. One of the important processing steps is the extraction of specific terms expressing risk indicators that can lead to the conclusion of HAI cases. This term extraction has to be very accurate and negation has to be taken into account in order to really understand if a string corresponding to a potential risk indicator is attested positively or negatively in the document. We propose a linguistically motivated approach for dealing with negation using both syntactic and semantic information. This approach is first described and then evaluated in the context of our application in the medical domain. The results of evaluation are also compared with other related approaches dealing with negation in medical texts.**

***Index terms*—Negation detection, discharge summaries, dependency parsing.**

I. Introduction

NEGATION is commonly used in natural language texts and is a challenge for general tasks of information extraction. In the medical domain, in particular, specific efforts for the annotation (see [13]), the description (see [10]), and the processing of negation (see [9] and [11]) have been made in the recent years. One evident application of processing negation is to make it possible to distinguish factual information from non-factual information expressed in the texts. This processing will benefit the classical tasks of information extraction like question-answering, summarization (where usually one wants to give priority to positive information). Furthermore, according to [8], explicit knowledge of what is negated may be also useful for a wide range of scenarios in the medical and biomedical domain.

In this paper, we present an algorithm which is able to detect negated information in French hospital discharge summaries. This work has been developed within a larger system detecting occurrences of hospital acquired infections (HAI) in texts. An overall presentation of the project can be found in [12]. One of the processing steps of the system is the extraction of terms and expressions which correspond to risk indicators for HAI. We want thus to be able to distinguish between factual and non-factual risk indicators. We focus in this paper on the negation processing task, which is integrated in the full system.

Manuscript received October 21, 2010. Manuscript accepted for publication January 19, 2011.

The author is with XRCE – Xerox Research Centre Europe, 6 Chemin de Maupertuis, 38240 Meylan, France. (e-mail: Caroline.Hagege@xrce.xerox.com).

II. Related Work

The abundant literature on the treatment of negation in medical and biomedical texts shows that this is a crucial problem. The BioScope corpus [13] is manually annotated with uncertain and negated information. This work reports that around 13% of sentences in the corpus contain negation. Existing systems dealing with negation in the medical domain use either machine learning algorithms as [9] or rule-based approaches. For all these systems the general goal is the same: finding negative triggers and their scope.

The Negfinder system presented in [11] is a rule-based system that first identifies medical terms, and then replaces them by an UMLS concept identifier. Then, a lexer and context-free negation grammars are applied. The output consists in the original text in which concepts and negation information is marked. [3] also presents a system based on regular expressions aiming at the detection of the presence or absence of a medical finding in texts. More recently, [6] describes a system in which negation together with temporality and experiencer contextual values are processed.

In all these approaches, the general method is to first define a set of trigger expressions. These expressions usually consist in a wide range of simple or complex lexicalized linguistic chunks that may induce a polarity change to the neighboring textual expressions. Then, once these expressions are found in text, a way to determine the scope of the negative trigger is proposed.

Another syntax-based method is presented in [4], where the authors describe the NegHunter system, which detects negation in Clinical Practice Guidelines. NegHunter considers

a more restricted and universal set of negative triggers compared with the previously cited approaches. As we will see in section 3.2, we adopt a similar approach for detecting negative triggers but with some differences that will be explained later.

Our method for dealing with negation is also rule-based and generalizes the approaches presented above making the following assumptions:

- Negative trigger expressions presented in the related literature can be generalized using parsing and some lexical semantics.
- Negative triggers should be general enough to be used for processing negation in different contexts and domains.
- Syntax is not enough to determine the scope of negation.

Take for instance the following expressions:

Absence d'origine évidente de cette septicémie.
(Absence of evident origin of this septicemia)
Absence de signes de septicémie.
(Absence of signs of septicemia)

Both expressions have the same syntactic structure: a nominal head *absence* followed by a modifying prepositional chunk headed by *origine* in the first expression and headed by *signes* in the second expression. These heads are in turn modified by another prepositional chunk headed by *septicémie* in both cases. We are interested in the fact that a patient has or has not septicemia. In the first case, the expression entails that the patient has septicemia and the negation carried out by *absence* indicates that the origin of septicemia is not clear. In the second case, the patient has no septicemia at all, and the negation carried out by *absence* has to be applied to the whole expression *signes de septicémie.* These examples show clearly the limits of a purely syntax-based approach for dealing with negation. In these examples syntactic structures are exactly the same and only the introduction of lexical semantics makes it possible to process these sentences in an appropriate manner. The need of a semantic processing of negation is also expressed in the approach presented in [4]. In this paper, the authors explain that their approach for dealing with negation is a first processing step which has to be completed with further semantic processing.

III. Negation Processing Methodology

This section details our negation processing methodology. Negation detection is integrated within a more general linguistic processor presented in [1], which deals with discharge summaries for the final purpose of HAI detection.

A. General Motivation

Our parser takes as input a text (discharge summary) and provides as output a linguistic representation of this text consisting in tokenization, part-of-speech tagging, chunking and the establishment of dependency links between the linguistic units of the text. Linguistic units consist in feature structures that carry morphological (inflectional), syntactic (part-of-speech, and some sub-categorization information) and some lexical semantic features. The parser also provides a Java API used for the implementation of extensions (as co-reference, temporal processing etc.).

For this specific task we use the French version of the parser that has been tuned for the processing of medical texts (introduction of dedicated medical lexicon and terminology, specific POS disambiguation rules for the medical domain). Our purpose is to detect automatically occurrences of hospital acquired infections in these texts, and one important step is to recognize in the texts risk indicators that may lead to the conclusion of an HAI. These risk indicators can be either medical terms or more complex expressions involving numerical values. Our linguistic processor uses a specialized lexicon (for simple terms) together with local and syntactic rules (for complex terms and numerical expressions) in order to mark all risk indicators belonging to the following classes (see [5] for more details):

- INFECTIOUS_DISEASE
  PRESENCE_OF_FEVER
  DIAGNOSIS
  VIRAL_DISEASE
  ANTIBIOTIC_ADMINISTRATION
  INTERVENTION
  PRESENCE_OF_INFECTIOUS_GERMS

For accurate detection, we want to be able to state if a textual occurrence of a risk indicator is negated or not. We will discard negative occurrences from the list of the potential risk indicators.

B. Negative Seeds

Related approaches dealing with negation usually have a first processing step consisting in the detection of what is usually called negative triggers. [10] describes negative cues found in the BioScope corpus. Negative triggers used by the NegEx algorithm presented in [3] and are publicly available. They consider expressions like *without any evidence of, without evidence, without indication of, without sign of* as negative triggers. If we examine these expressions carefully, we can see that they correspond to the following pattern: they are prepositional phrases (PP) introduced by *without* and the nominal head of these PP correspond to one of the nouns *evidence, indication,* or *sign.* All these nouns are in the same semantic field (for instance in the synonym dictionary available on-line at http://dico.isc.cnrs.fr/dico/en/search).

Instead of considering negative triggers, we decided to consider only what we call negative seeds. Negative seeds consist in a small set of linguistic units, which have the property of negating the syntactic heads they are linked to. The negative seeds are very general and universal and can be used as negation introductors for all kinds of documents and domains. We differ here from [4] (which also considers more general negative triggers) by not considering any verb or noun like *absence* for which negation is induced by the lexical semantics attached to the word. The list of our 14 negative

seeds is given here exhaustively. Negative seeds are presented according their distributional properties.

1. Determiners (*aucun, ni, pas de, point de, nul*) which negate the nominal head they determine (e.g. *aucune* infection). This corresponds to the *DETERM* dependency calculated by the parser
2. Adjectives (*nul, inexistant, négatif*) which either negate the nominal head they qualify (e.g. infection *inexistante*) or, when they are used as subject complements, they negate the subject of the copulative verb (e.g. l'infection est *inexistante*). This corresponds to the *NMOD_POSIT1* dependency calculated by the parser.
3. Discontinuous negation adverbs (*ne ...pas, ne...aucun, ne...point, ne...plus*) which negate the verbal predicate situated either in the discontinuous part of these adverbs (for simple forms) or on the right of the adverb (for participial verbal forms). This corresponds to a *VMOD_POSIT1* dependency calculated by the parser. A restriction is added in order to avoid taking as negative seeds these adverbs when they are modified by other adverbs as *presque, quasiment* (almost). For instance, *Il **n'**a presque **pas** de fièvre* means *he has almost no fever.*
4. A simple adverb (*non)* which negates its head. This head can be an adjective, a past participle and sometimes a noun.
5. A preposition (*sans*) which always negates the nominal head of the prepositional phrase they introduce. This corresponds to the *PREPD* dependency calculated by the parser.

All these linguistic elements change the polarity of the syntactic head that is in a direct dependency relation with them.

For instance, in the following expressions negative triggers are indicated in bold and negated syntactic heads are underlined.

*Le patient **n'a pas** présenté de fièvre*
(The patient did not show any fever)
***Aucun** signe d'infection à ce jour.*
(No sign of infection this day)

Concretely, during parsing, these negative seeds create a unary relation *NEGAT* on the verbal or nominal head associated to them. Taking the two examples mentioned above, two unary relations are thus calculated: *NEGAT(présenté) and NEGAT(signe).*

It is important to note, that since we work with dependency relations, the fact that the negated head is on the right or on the left of the negative seed is not a concern for us.

C. Semantic Fields and Their Polarity

We also consider a subset of lexical units belonging to specific semantic fields. As our final purpose is to be able to distinguish if a term mentioned in text is attested or not attested, we are interested in words belonging to semantic fields denoting the existence, the evidence, the continuation of a fact or an event. More specifically, we consider the following semantic fields:

- existence/non existence
  evidence/non evidence
  continuity/break
  augmentation/diminution

Nouns and verbs belonging to these fields will have an a-priori polarity associated to them. Intuitively, a noun stating the existence a fact (like *sign* or *existence)* will have a positive polarity, and on the contrary, a noun like *absence* will carry a negative polarity. The collection of lexical units belonging to these fields has been compiled using the online synonym dictionary for French developed by the Caen University (http://www.crisco.unicaen.fr/cgi-bin/cherches.cgi). We established a list of 122 verbs and nouns. They are coded in the lexicon of our linguistic processor using Boolean features corresponding to the above mentioned semantic fields.

For instance, the verb *attester (attest)* and the noun *preuve (proof)* belong to the *evidence* semantic field. As such, they have the boolean feature *[evidence:+]* associated to the corresponding lexical entries. The noun *persistance (persistence)* is of *continuity* semantic field and bears the feature *[continuity:+]*, and the noun *fin (end)* corresponds to a noun of non-continuity semantic field bearing thus the feature *[continuity:-]*. Note that these semantic fields may be only relevant to one specific reading of a semantically ambiguous lexical unit. However, because we deal with a specific domain, semantic ambiguity is here limited.

We can in our linguistic processor generalize over features carried by the lexical entries. For instance, we can state that any feature *[evidence:+]* implies a feature *[polarity:+]*. These kinds of generalizations are performed in configuration files read by the parser. As a result, all the lexical entries coded with the above-mentioned features related with semantic fields will have an associated *polarity* feature which can have the value + (for positive polarity) or the value – (for negative polarity).

We can then propagate polarities in order to finally detect what is negated or not.

D. Polarity Propagation

Syntactic negation (expressed by the *NEGAT*relation introduced by negation seeds) and a-priori polarities of lexical heads are then combined in order to propagate negative/positive polarity information from one head to its complement.

Two simple rules for polarity propagation are used:

1. If a linguistic head has a NEGAT and if it has an a-priori polarity associated, then this polarity is inverted.
2. If a linguistic head has no NEGAT relation but bears information on polarity and if its modifier also has an explicit polarity, then polarities are combined (as it is explained later), and a new polarity is given to the modifier.

1) Polarity Inversion due to Syntactic Negation

The first rule corresponds to the fact that a syntactic negation marker inverts the polarity of the negated lexical unit. It can be illustrated by the following expression: *aucun signe d'infection* (*no sign of infection*).

The negative seed *aucun* has created a NEGAT relation on the word *signe*. This word has a feature *[polarity:+]* since it is a noun of the semantic field *evidence*. In this specific context, the polarity will be inverted and the new value of the feature *polarity* will become -.

In a similar way for the word *absence* in the context *pas d'absence de signe d'infection*, the negation seed *pas de* creates a NEGAT relation to the word *absence* which has an a-priori *[polarity:-]* feature. As a result, in this specific context, as polarity will be inverted, *absence* will finally bear the feature *[polarity:+]*

The following statement is added in the grammar rule files read by the parser. This statement says that anything having the feature *[polarity:+]* (expressed by the first line of the rule) and NEGAT relation (first condition in the second line), will have first the feature polarity suppressed (expressed the second condition *#1[polarity=~]*) and then the feature polarity is set to – (expressed by the last condition of the expression *#1[polarity=-]*). A similar statement changing *[polarity:-]* into *[polarity:+]* for negated lexical heads is also present in the grammar files.

```
| #1[polarity:+] |
if ( NEGAT(#1) & #1[polarity=~] & #1[polarity=-] ) ~
```

2) Polarity Combination from Head to Modifier

Rule 2) mentioned above expresses the fact that the polarity carried by a syntactic head may have influence on the polarity of its complement. Intuitively, in an expression like *lack of food*, where *lack* is the syntactic head and *food* the complement, the final status concerning the existence or nonexistence of *food* is ruled out by the fact that *lack* introduces semantically the idea of absence.

Polarity propagation is implemented taking advantage of the general syntactic dependencies computed by the parser.

Two possibilities can occur:

1. The argument or modifier of a lexical head with a polarity also has an a-priori polarity. In this case the polarity of the argument/modifier is changed according to table 1. Polarity propagation will then once again be applied on this argument/modifier.
2. The argument or modifier of a lexical head with a polarity has no a-priori polarity. In this case, the argument/modifier will be negated if the polarity of the lexical head is – or not negated if the polarity of the lexical head is +. Polarity propagation stops on this modifier.

Note that there is an order in the choice of arguments/modifiers for polarity propagation:

– Arguments are taken before modifiers (which means that for a verb, its object complement will be considered before any kind of modifying PP).

In case of multiple modifier choice, the left-most modifier is chosen.

Returning to the example *il n'y a pas d'absence de signe d'infection*, the syntactic negation of the verb *avoir* creates a *NEGAT(a)* relation. As *absence* has an a-priori polarity *[polarity:-]*, the initial polarity attached to *absence* is changed and becomes *[polarity:+]*. The word *absence* is in turn modified by the word *signe* which has an a-priori polarity set to +. The combination of both *[polarity:+]* gives a final *[polarity:+]* to the word *signe* (according to Table 1). Finally, *infection*, which has no a-priori polarity, modifies *signe*. It is not negated because it modifies a lexical unit with *[polarity:+]* feature.

TABLE I
POLARITY COMBINATION

HEAD MODIFIER	Polarity:+	Polarity:-
Polarity:+	Polarity:+	Polarity:-
Polarity:-	Polarity:-	Polarity:+

E. Negation Focus

Negation focus is the final unary relation that is established when polarity propagation stops.

This propagation stops in two situations:

1. A lexical unit with a polarity has no complements or modifiers.

The complement or modifier of the lexical unit bearing a polarity has no a-priori polarity associated to it.

The first case can be illustrated by the following example:

*Il **n**'y a **pas** d'augmentation.*
(There is no increase)

The verb *a* bears a negative polarity because of the syntactic negation. Its complement *augmentation* has an a-priori positive polarity and receives a negative polarity during the polarity propagation process. Propagation then stops in the absence of any modifier of *augmentation*. In this case, because *augmentation* has a negative polarity which cannot be propagated, it will correspond to what we call the negation focus. The parser produces a unary dependency *NON(augmentation)*.

2. The second case can be illustrated as follows:

On ne retrouve pas d'infection suite aux examens.
(No infection was found after the examinations).

The verb *retrouver* receives a negative polarity as it is involved in a unary *NEGAT* relation created by the negation seed *ne...pas*. The word *infection* has no a-priori polarity associated to it but it is a complement of *retrouver [polarity:-]*. The propagation stops on the word *infection* which is also

the focus of the negation. A unary dependency *NON(infection)* is also created by the parser.

An important issue for polarity propagation and negation focus detection is the fact that an accurate PP attachment is necessary in order to get good results. For instance in a case like *Il n'y a pas de suspicion depuis la semaine dernière. (There has been no suspicion since last week)*, the temporal PP is attached to the main verb *avoir* and not to the noun *suspicion*. If an error of PP attachment occurs, polarity propagation would be wrong and the final negated element would be *semaine* and not *suspicion*.

IV. Experiment and Evaluation

In order to test our approach, we first trained the system in the following way. We perform two runs of the same set of texts.

- The first run extracts all risk indicators without using any information regarding negation (we disabled the lexical enrichment and the grammar rules for negation propagation). As a result, any occurrence of risk indicators is extracted regardless of the fact if they are negated or not.
- The second run uses the same system but enriched with negation processing. In this case, only non negated risk indicators (according to our system) are extracted.

The outputs of these two runs contain the initial text with the risk indicators annotated and colored. The two files are aligned and compared. Any difference between the two runs are examined and verified. During the training phase, we add some extra lexical entries, consider new negation seeds, and tune some rules.

After training, we perform a test in order to evaluate the accuracy in detecting negative risk indicators in our discharge summaries. We took a set of 110 unseen discharge summaries coming from different hospitals and different care units 42) documents for an intensive care unit, 50 documents for .(orthopedics and 18 documents for digestive surgery

These documents were first processed using the system without negation processing in order to detect and mark all possible occurrences of HAI risk indicators. All the marked occurrences were then verified manually and the annotator decided for each of them if they were negated or not. As we only treated negation and not modality, we consider that uncertainty is to be annotated as positive and not as negative. Furthermore, we did not take into account temporality. As a result, any mention of a future or past occurrence of a risk indicator is considered as positive if it is positively stated. We then process automatically the same documents with our system enriched with negation processing and compare the automatic and manual annotations. We obtained the following results:

TABLE II
Experimental results

	Manually annotated negative risk indicators	Manually annotated positive risk indicators
System annotated negative risk indicators	True positives (TP) 174	False positives (FP) 8
System annotated positive risk indicators	False negatives (FN) 6	True negatives (TN) 2,255

Precision, recall, specificity and accuracy are then calculated. We obtained the following figures:

Precision: TP/(TP+FP) 95.6%
Recall/Sensitivity = TP/(TP+FN) = 96.6%
Specificity = TN/(TN+FP) = 99.6%
Accuracy = (TP+TN)/(TP+TN+FP+FN) = 99.4%

These figures show that we obtain very good results. Compared with [11] which performs a comparable evaluation (verification of negated terms) our results are higher (specificity obtained was 97.7% and sensitivity was 95.3%). However, we do not know exactly the kind of texts that were processed in [11], and we only consider a subset of terms which may lead to less variety in expressing negation.

V. Discussion

The method we present shows to be very effective for detecting negative terms on the kind of medical documents we processed (French discharge summaries). The good results we obtain is explained by the fact that we make use of both syntactic and semantic information. Furthermore, because our underlying syntactic knowledge is expressed in terms of dependencies, the distance between words for finding the negation scope is not a concern. Our algorithm is completely integrated within our general purpose linguistic processor. However, the approach is easily adaptable to any other dependency parser. One of the advantages of our method is that it treats syntactic and lexically induced negation. Double negation, although not very frequent in medical narratives is processed naturally and straightforwardly, which is not the case in related approaches ([2] states in the discussion section that double negation is a problem for their system).

However, in this work we restricted the analysis to simple negation (negative conditional expressed by expressions like *either...or*, and uncertainty introduced by tense and modality are not considered). Examples of these more complex negation cases can be found in [10] and it would be interesting to enhance our system in order to take them into account.

Since we use an existing dependency parser and since the specific lexical and syntactic coding is very limited (addition of features on approximately 120 words and 6 additional rules in our grammar), this approach is easily portable for other languages for which we have a dependency parser[1]. Our results are however very dependent on the parser accuracy. PP attachment is one of the key issues, and it may lead to

[1]At least for all romance languages, English and German.

erroneous polarity propagation. Part-of-Speech disambiguation errors may also be a problem as they impact the computing of the dependencies used for polarity propagation. Furthermore, lexical semantic ambiguity can also be a concern if we enlarge this approach to other domains. This kind of ambiguity can lead to erroneous attribution of a-priori polarities, which will impact the correct computing of the negated element.

VI. Conclusion

We have presented a method for dealing with negation in unstructured medical discharge summaries written in French. The method we propose makes use of both syntactic and semantic information and is integrated within a larger linguistic processor for unstructured texts. This approach is suitable to other languages and should be easily adaptable, as coding effort to integrate negation processing in the parser is limited. One of the advantages of our approach is that it treats in a homogeneous way negation expressed syntactically and negation induced lexically.

The next step will be to test this approach to medical texts that are not discharge summaries and even to texts in other domains. We believe that we can extend this approach to other domain-dependent texts (possibly with some changes in the lexical coding). We also would like to apply this approach to the treatment of English medical texts in order to take advantage of already existing annotated resources for the evaluation and comparison of the results with other existing systems. Finally, we would like to enlarge negation detection to a more general system of factuality detection, which will take into account modality, conditionality and uncertainty.

References

[1] S. Aït-Mokhtar, J.P. Chanod, and C. Roux, "Robustness beyond Shallowness: Incremental Deep Parsing," *Natural Language Engineering*, 8, pp.121-144, 2002.

[2] P.L. Bikin, S.H. Brown, B.A. Bauer, C.S. Husser, W. Carruth, L.R Bergstrom, and D.L. Wahner-Roedler, "A controlled trial of automated classification of negation from clinical notes," *BMC Medical Informatics and Decision Making*, S:13, 2005.

[3] W.W Chapman, W. Bridewell, P. Habury, G.F. Cooper, and B.G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of Biomedical Informatics*, 34, pp. 301-310, 2001.

[4] S. Gindl, K. Kaiser, and S. Miksh, "Syntactical Negation Detection in Clinical Practice Guidelines," in Andersen, S.K.; Klein, G.O.; Schulz, S.; Aarts, J.; Mazzoleni, M.C. (eds.) *eHealth Beyond the Horizon – Get IT There. Proc. of the 21st International Congress of the European Federation for Medical Informatics (MIE 2008)*, Göteborg, Sweden, IOS Press, 2008, pp. 187-192.

[5] C. Hagège, P. Marchal, Q. Gicquel, S. Darmoni, S. Pereira, M-H. Metzger, "Linguistic and Temporal Processing for Discovering Hospital Acquired Infection from Patient Records," in *Proceedings of the 2nd International Workshop on Knowledge Representation for Health Care (KR4HC-2010)*, Lisbon, Portugal, 2010.

[6] H. Harkema, J. N. Dowling, T. Thomblade, and W. Chapman, "ConText: An Algorithm For Determining Negation, Experiencer, and Temporal Status from Clinical Reports," *Journal of Biomedical Informatics*, 42, pp. 839-851, 2009.

[7] Y. Huang and H.J. Lowe, "A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 14(3), pp. 304-311, 2007.

[8] Y. Krallinger, "Importance of negations and experimental qualifiers in biomedical literature," in *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, Uppsala, Sweden, 2010, pp. 46-49.

[9] R. Morante and W. Daelemans, "A metalearning approach to processing the scope of negation," in *Proceedings of CoNLL 2009*, 2009, pp. 21-29.

[10] R. Morante, "Descriptive analysis of negation cues in biomedical texts," in *Proceedings of the Seventh International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010, pp. 1429-1436.

[11] P.G. Mutalik, A. Deshpande, and P.M. Nadkarni, "Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study Using the UMLS," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 8(6), pp. 598-609, 2001.

[12] D. Proux, P. Marchal, F. Segond, I. Kergoulay, S. Darmoni, S. Pereira, Q. Gicquel, M-H. Metzger, "Natural Language Processing to detect Risk Patterns related to Hospital Acquired Infections," in *Proceedings of RANLP 2009*, Borovetz, Bulgaria, 2009, pp. 865-881.

[13] V. Vincze, G. Szarvas, R. Farkas, G. Móra, and J. Csirik, "The Bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes," *BMC Bioinformatics*, 9(Suppl 11):S9, 2008.

A Micro Artificial Immune System

Juan Carlos Herrera-Lozada, Hiram Calvo, and Hind Taud

Abstract—**In this paper, we present a new algorithm, namely, a micro artificial immune system (Micro-AIS) based on the Clonal Selection Theory for solving numerical optimization problems. For our study, we consider the algorithm CLONALG, a widely used artificial immune system. During the process of cloning, CLONALG greatly increases the size of its population. We propose a version with reduced population. Our hypothesis is that reducing the number of individuals in a population will decrease the number of evaluations of the objective function, increasing the speed of convergence and reducing the use of data memory. Our proposal uses a population of 5 individuals (antibodies), from which only 15 clones are obtained. In the maturation stage of the clones, two simple and fast mutation operators are used in a nominal convergence that works together with a reinitialization process to preserve the diversity. To validate our algorithm, we use a set of test functions taken from the specialized literature to compare our approach with the standard version of CLONALG. The same method can be applied in many other problems, for example, in text processing.**

Index terms—**Artificial immune system, Clonal selection theory, micro algorithm, numerical optimization.**

I. Introduction

BIO-INSPIRED and Evolutionary algorithms have become very important within the area of artificial intelligence, because they have proved successful in solving certain complex problems of machine learning, classification and numerical optimization [1]. Such techniques are population based, in other words, they use a population of potential solutions enabling a wide exploration of the search space. The simplicity of an algorithm is one of the current trends in the field of evolutionary computation, although in the majority of cases, the performance is sacrificed in favor of a lower computational cost [2]. Due to the simultaneous manipulation of a large set of solutions, implementation of an algorithm requires a large space in data memory and generally high processing time. To reduce these factors, algorithms with extremely small populations are designed, and for most applications their performance is comparable with the standard population algorithms [3].

As the evolutionary algorithms, Artificial Immune System (AIS) has been successfully applied to a variety of optimization problems [4]. AIS is a computational intelligence paradigm inspired by the biological immune system which has found application in pattern recognition and machine learning. Different ways of AIS for optimization as the immune network theory and the clonal selection principle have been proposed and implemented by different researchers as explained in [5].

The main motivation of our research is to propose a simple and powerful algorithm which presents a reduced computational cost when using a micro population of individuals within a clonal proliferation scheme which is the central point in the functioning of artificial immune system.

Two novel mutation operators were designed and implemented. These operators accelerate the convergence by providing a uniform search to avoid getting into local optimum.

In this work we apply micro-AIS for numerical optimization, but the same method can be applied in many other problems, for example, in text processing. It is promising to apply bio-inspired algorithms (more specifically, genetic algorithms) in text processing tasks, see, for example, [6].

A. Previous Work

In [3] Goldberg introduced the concept of nominal convergence when he experimented with a simple genetic algorithm (GA) using a population of only 3 individuals. He found that these 3 chromosomes were sufficient to ensure convergence of the algorithm regardless of the size of them, aided by a process of elitism. Goldberg applied genetic operators in a nominal convergence which is controlled by two possible parameters: a specified number of generations or a degree of similarity among all chromosomes. At the end of the nominal convergence, the best individual is preserved and two individuals are randomly generated: they will form the new population.

In [7] Krishnakumar designed a GA with a population of 5 individuals and he named his algorithm *Micro Genetic Algorithm* (Micro-GA). Like Goldberg, Krishnakumar used elitism to preserve the best single strand found at the end of nominal convergence, as one of the individuals used for the next generation. When comparing the performance of the Micro-GA with a simple GA with a population of 50 individuals, better results were obtained on functions of only one objective and the GA with a reduced population converged faster. Krishnakumar's algorithm has achieved good results when it is used to solve optimization problems for high-dimensional functions [8].

Dozier *et al*. in [9] presented two heuristic-based micro genetic algorithms which quickly find solutions to constraints satisfaction problem. They experimented with different sizes of micro population and found that for a particular problem, a relatively small number of individuals in the genetic algorithm was sufficient.

Manuscript received March 12, 2011. Manuscript accepted for publication June 6, 2011.

Juan Carlos Herrera-Lozada and Hiram Calvo are with Centro de Investigación en Computación, Instituto Politécnico Nacional, México D. F., 07738, Mexico (e-mail: jlozada@ipn.mx, hcalvo@cic.ipn.mx).

Hind Taud is with Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, México D. F., 07700, Mexico (e-mail: htaud@ipn.mx).

Coello and Toscano designed a Micro-GA for solving the multi-objective optimization problem [10], providing criteria for the management of constraints, besides proposing a scheme of Pareto dominance with a geographical location to maintain the diversity and uniformly distribution of the solutions on the Pareto front. This algorithm works with a population of 4 individuals and uses a secondary memory that stores potential solutions throughout the search. This approach was widely used to successfully solve various engineering problems as discussed in [11] and [12].

Recently, Fuentes and Coello in [13] designed a micro algorithm for PSO (Particle Swarm Optimization) to solve optimization problems of one objective and constraints satisfaction. They use 5 particles (individuals) helped by a nominal convergence.

With regard to artificial immune systems with small population, there are no studies reported in the literature. There are, however, certain similarities with the works cited above:

1. Population size of 3 to 5 individuals.
2. Nominal convergence is required as well as a reinitialization process.
3. Elitism is necessary to preserve at least the best individual obtained at the end of the nominal convergence.

II. Artificial Immune System

De Castro and Von Zuben developed the Clonal Selection Algorithm (CLONALG) on the basis of clonal selection theory of the immune system [14, 15]. Clonal Selection is based on the way in which both B-cells and T-cells adapt in order to match and kill the foreign cells. This algorithm can perform pattern recognition and adapt to solve multimodal optimization tasks. The block diagram of CLONALG is shown in Fig. 1. This algorithm is described as follows:

(1) Generate (randomly) a set (P) of candidate solutions or antibodies, composed of the memory cells (M) and the remaining population (Pr), ($P = Pr + M$);

(2) Select the n best antibodies (P_n), based on affinity;

(3) Clone these n best antibodies in proportion to their affinity using $N_c = \sum_{i=1}^{n} round\left(\frac{\beta \cdot N}{i}\right)$ where N_c is the total number of clones generated for each of the antigens like objective function, β is a multiplying factor, N is the total number of antibodies, and $round(.)$ is the operator that rounds its argument toward the closest integer. Each term of this sum corresponds to the clone size of each selected antibody, e.g., for N=100 and $\beta = 1$, the antibody with highest affinity will produce 100 clones; the antibody with the second highest affinity produces 50 clones, and so on, giving rise to a temporary set of clones (C);

(4) Apply a hypermutation to the temporary clones. The degree of mutation is inversely proportional to the affinity. The maturated antibodies are generated (C*);

(5) Re-select the best elements from C* to compose the memory set M. Some members of P can be replaced by other improved members of C*;

(6) Replace d antibodies by novel ones to introduce the diversity concept. The probability to be replaced is inversely proportional to the affinity of the previous remaining population (Pr).

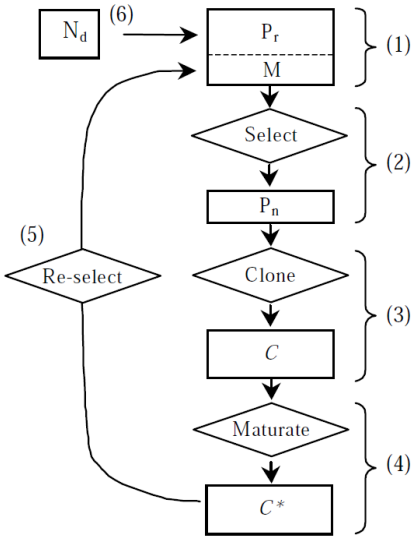


Fig. 1. Block diagram of the clonal selection algorithm CLONALG by De Castro and Von Zuben.

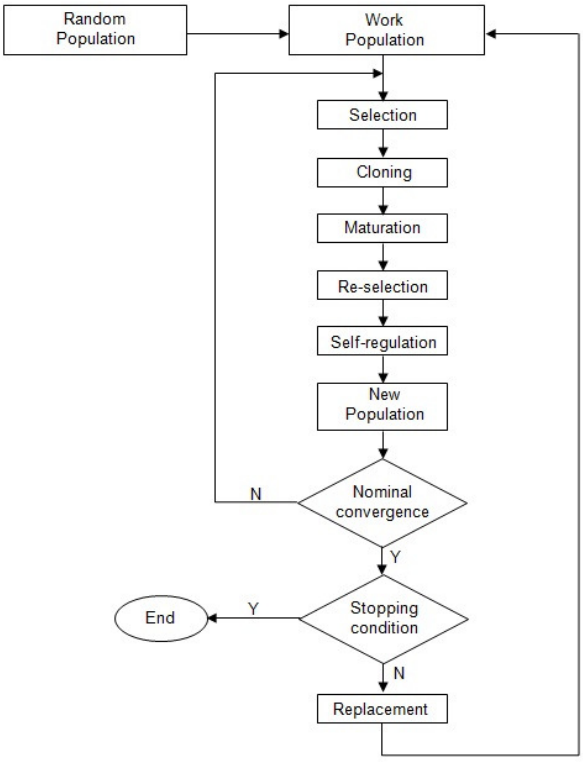


Fig. 2. Micro-AIS.

III. Micro Artificial Immune System

Fig. 2 shows our algorithm. Our methodology is based on the methodology proposed by Goldberg in [3]: the variation operators are applied to a small population (randomly generated) to achieve nominal convergence. Subsequently, a new population should be generated by transferring the best individuals of the population obtained after the convergence to the new one. The remaining individuals are randomly generated.

The proposed algorithm works as follows:

(1) Generate randomly a population of 5 antibodies (individuals). In the initial generation, these antibodies are copied directly to the working population and nominal convergence is controlled by the number of generations, in our case equal to 10.

(2) Use selection based on ranking. The antibody with the highest affinity will be the best individual. In our algorithm we named this individual as *BestAb*.

(3) Perform the cloning of the antibodies using $N_C = \sum_{i=1}^{n} (n - (i-1))$, where N_C is the number of clones to be generated for each antibody, n is the total number of antibodies of the population and i is the current antibody starting from the antibody with the highest affinity (*BestAb*).

(4) Consider a population of 5 antibodies and generate a population of 15 clones: *BestAb* antibody gets 5 clones; the second ranking antibody gets 4 clones and so on until the worst antibody that gets a single clone.

(5) Perform the maturation of clones using mutation process. The probability of mutation is set at the beginning of nominal convergence for each group of clones obtained from the same antibody. This probability is determined in proportion to the affinity of each antibody and decreases uniformly in each generation, so the group of clones obtained from *BestAb* mutates less than other groups of clones that have been generated from the remaining antibodies. The single clone that we got from the worst antibody has the highest possibility to mutate. For this purpose we use

$$prob\_mutation(i) = \frac{Aff(i)}{\sum_{i=1}^{n} Aff(i)}$$

where i is the antibody that will set the mutation probability for the group of clones that were obtained from himself and n is the total antibody population. To decrease the mutation probability uniformly in each generation, within the nominal convergence we used

$$if\ prob \leq \frac{prob\_mutation(i)}{generation}\ then\ to\ apply\ mutation$$

where $random\ prob \in [0,1]$ and *generation* variable is the current generation within nominal convergence considering $int\ generation \in [1,10]$. Note that we should not divide by zero.

For the variation of each of the clones, we present two operators that are rather simple and mostly exploit the search space to perform different step sizes in the process of mutation. Several aspects have been considered to implement these operators: the number of clones, the current generation within nominal convergence and the permissible range of values of the decision variables. We use the following two mutation operators, with a 50% probability, which act on each decision variable of a clone (in our scheme, the entire solution vector is mutated):

$$x' = x + \frac{(\alpha \cdot range \cdot generation)}{N_C}$$

and

$$x' = x + \frac{(\alpha \cdot range)}{(generation \cdot N_C)}$$

where x' is the mutated decision variable, x is the decision variable to mutate, α is a uniform random number where $random\ \alpha \in [0,1]$, *generation* is the current generation within the nominal convergence and N_c is the total number of clones. The value of α is computed for each decision variable of the clone.

In case of the 5 clones derived from *BestAb*, $range \in [LB, UB]$ is a random number between the lower bound (LB) and the upper bound (UB) of decision variables and it is a constant value for all the dimension of the clone, in other words, it has the same value for all decision variables of the clone.

For the remaining clones which were obtained from the other 4 antibodies, *range* is any value (decision variable) from *BestAb* antibody which is chosen randomly.

The first operator using in the mutation generates step sizes larger than the second operator.

(6) Make another selection based on ranking. This time, we sort the 15 clones with respect to their affinity. We must select the two best clones (elitism) and the new population is completed with 3 other clones selected randomly from the population of mature clones. The remaining clones will be eliminated, providing a self-regulation within the nominal convergence.

(7) When nominal convergence is achieved (while working with 10 generations), we keep the two best clones, and other 3 antibodies are generated randomly to complete the new working population and the nominal convergence starts again until the algorithm achieves the stop condition.

IV. Experimental Setup

In order to validate the proposed approach, we used the multivariate functions presented in [16]. These functions are listed in Appendix A. All selected test functions have 30 variables (dimensions) and an optimum value at zero, except for *f08* with an optimum at -12569.5. For all cases we used a population of 5 individuals and nominal convergence in 10 generations. The general stop criterion of the algorithm varied depending on the problem to be solved. For the experiments, we used a 2.66 GHz Quad Core PC with 2MB. Table I shows the results for 20 runs of the algorithm.

TABLE I
RESULTS OBTAINED WITH MICRO-AIS

Function	External cycle	Nominal Convergence	Best	Worst	Mean
f01	1000	10	0.0	0.000022	0.000009
f02	1000	10	0.0	0.000017	0.000008
f03	1000	10	0.0	0.000002	0.000001
f04	1000	10	0.0	0.000012	0.000005
f05	1000	10	0.0	0.000028	0.000012
f06	2000	10	0.0	0.000032	0.000015
f07	2000	10	0.0	0.000027	0.000013
f08	2000	10	-12569.5	-12569.57	-12569.496
f09	2000	10	0.0	0.000033	0.000013
f10	2000	10	0.0	0.000011	0.000007
f11	2000	10	0.0	0.000013	0.000004

TABLE II
CLONALG VS. MICRO-AIS

	Ab (antibodies)	Clones	Nominal Convergence	External cycle	Evaluations to objective function	Time (seconds)
CLONALG						
f01	50	256	0	1000	1,280,000	47.2
f05	70	312	0	1000	21,840,000	78.6
f07	70	312	0	1200	26,208,000	103.7
Micro-AIS						
f01	5	15	10	1000	750,000	14.8
f05	5	15	10	1000	750,000	14.2
f07	5	15	10	2000	1,500,000	48.3

To validate the performance of our algorithm with respect to the standard version of CLONALG, we compared it with some of the above mentioned functions under equal conditions. The main results are related with the number of evaluations of the objective function and convergence time. Table II lists these results for 20 runs of both algorithms. We implemented the adaptations to CLONALG for using multivariate functions. For CLONALG we used a multiplication factor $\beta = 1$ and the number of antibodies listed in Table II.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a new micro algorithm based on clonal selection theory for solving numerical optimization problem. Since the model of the artificial immune system does not include a crossover operator, cloning (set to 15 clones in our case) and mutation represent the main challenges for maintaining diversity.

Two mutation operators were designed in our approach showed excellent solutions with a low computational cost. These operators were used without modifications in all selected test functions. As shown in the results listed in Tables I and II, the Micro-AIS converges faster than CLONALG and uses less data memory. The nominal convergence and elitism of 40% of the population (considering only 5 antibodies) are of great importance to ensure the proper functioning of the algorithm.

Future work is aimed at the following four aspects:

- Find faster mutation operators,
- Design versions for handling constraints and muti-objective optimization,
- Develop possible hardware architectures, and
- Develop applications in different areas (for example, in text processing) and experiment with them.

REFERENCES

[1] D. Ashlock, *Evolutionary Computation for Modeling and Optimization*, Springer, 2005.

[2] M. Munetomo and Y. Satake, "Enhancing Model-building Efficiency in Extended Compact Genetic Algorithms," in *ICSMC '06. IEEE International Conference on Systems, Man and Cybernetics, 2006*, Volume 3, Oct. 8-11, 2006, pp. 2362-2367.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, 1989.

[4] L. Nunes de Castro and J. Timmis, *Artificial Immune Systems: A new Computational Intelligence Approach*, Springer, 2002.

[5] D. Dasgupta, "Advances in artificial immune systems," *Computational Intelligence Magazine*, IEEE, vol 1, issue 4, pp. 40-49, Nov. 2006.

[6] A. Gelbukh, G. Sidorov, D. Lara-Reyes, and L. Chanona-Hernandez, "Division of Spanish Words into Morphemes with a Genetic Algorithm," *Lecture Notes in Computer Science*, 5039, Springer-Verlag, pp. 19-26, 2008.

[7] K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization" in *SPIE Proceedings: Intelligent Control and Adaptive systems*, 1989, pp. 289-296.

[8] G. Alvarez, *Can we make genetic algorithms work in high-dimensionality problems?* Stanford Exploration Project (SEP) report 112, 2002.

[9] G. Dozier, J. Bowen and D. Bahler, "Solving Small and Large Scale Constraint Satisfaction Problems Using a Heuristic-Based Microgenetic Algorithm," in *Proceedings of the First IEEE Conference on Evolutionary Computation (ICEC'94),* Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel and H. Kitano (eds), 1994, pp. 306-311.

[10] G. Toscano, Carlos. A. Coello, "A Micro-Genetic Algorithm for multiobjective optimization," in *First International Conference on*

Evolutionary Multi-criterion Optimization, Lecture Notes in Computer Science, vol. 1993, Springer, 2001, pp. 126-140.

[11] Y. Ming and L. Cheng, "Application of Micro Genetic Algorithm to Optimization of Time-Domain Ultra-Wide Band Antenna Array," in *Microwave and Millimeter Wave Technology, 2007, ICMMT '07, International Conference*, April 2007, pp. 1-4.

[12] J. Mendoza, D. Morales, R. López, J. Vannier, and C. A. Coello, "Multiobjective Location of Automatic Voltage Regulators in a Radial Distribution Network Using a Micro Genetic Algorithm," *IEEE Transactions on Power Systems*, vol. 22, issue 1, pp. 404-412, Feb. 2007.

[13] J. C. Fuentes and C. A. Coello, "Handling Constraints in Particle Swarm Optimization Using a Small Population Size," *Lecture Notes in Computer Science, MICAI 2007: Advances in Artificial Intelligence*, vol. 4827, Springer, 2007.

[14] L. Nunes de Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of Genetic and Evolutionary Computation Conference, Workshop on AISAA*, July 2000, pp. 36-37.

[15] L. Nunes de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Trans. Evol. Comput.*, vol. 6, no. 3, pp. 239-251, Jun. 2002.

[16] E. Mezura, J. Velázquez, and C. A. Coello, "A comparative study of differential evolution variants for global optimization," in *ACM, GECCO 2006*, pp. 485-492.

APPENDIX A

Multivariate functions for the experimental setup, taken from [15].

f01 – Sphere Model

$$f_1(x) = \sum_{i=1}^{30} (x_i)^2$$

$-100 \leq x_i \leq 100$

$min\ (f_1) = f_1\ (0, \ldots, 0) = 0$

f02 – Schwefel's Problem

$$f_2(x) = \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i|$$

$-10 \leq x_i \leq 10$

$min\ (f_2) = f_2\ (0, \ldots, 0) = 0$

f03 – Schwefel's Problem

$$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$$

$-100 \leq x_i \leq 100$

$min\ (f_3) = f_3\ (0, \ldots, 0) = 0$

f04 – Schwefel's Problem

$$f_4(x) = max_i \{ [x_i], 1 \leq i \leq 30 \}$$

$-100 \leq x_i \leq 100$

$min\ (f_4) = f_4\ (0, \ldots, 0) = 0$

f05 – Generalized Rosenbrock's Function

$$f_5(x) = \sum_{i=1}^{29} \left| 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right|$$

$-30 \leq x_i \leq 30$

$min\ (f_5) = f_5\ (1, \ldots, 1) = 0$

f06 – Step Function

$$f_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2$$

$-100 \leq x_i \leq 100$

$min\ (f_6) = f_6\ (0, \ldots, 0) = 0$

f07 – Quartic Function with Noise

$$f_7(x) = \sum_{i=1}^{30} i x_i^4 + random[0,1)$$

$-1.28 \leq x_i \leq 1.28$

$min\ (f_7) = f_7\ (0, \ldots, 0) = 0$

f08 – Generalized Schwefel's Problem

$$f_8(x) = \sum_{i=1}^{30} \left(x_i \sin\left(\sqrt{|x_i|} \right) \right)$$

$-500 \leq x_i \leq 500$

$min\ (f_8) = f_8\ (420.9687, \ldots, 420.9687) = -12596.5$

f09 – Generalized Rastrigin's Problem

$$f_9(x) = \sum_{i=1}^{30} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$$

$-5.12 \leq x_i \leq 5.12$

$min\ (f_9) = f_9\ (0, \ldots, 0) = 0$

f10 – Ackley's Function

$$f_{10}(x) = -20e\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2} \right) - e\left(\frac{1}{30}\sum_{i=1}^{30} \cos(2\pi x_i) \right) + 20 + e$$

$-32 \leq x_i \leq 32$

$min\ (f_{10}) = f_{10}\ (0, \ldots, 0) = 0$

f11 – Generalized Griewank's Function

$$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$-600 \leq x_i \leq 600$

$min\ (f_{11}) = f_{11}\ (0, \ldots, 0) = 0$

A Graph-based Approach to Cross-language Multi-document Summarization

Florian Boudin, Stéphane Huet, and Juan-Manuel Torres-Moreno

***Abstract*—Cross-language summarization is the task of generating a summary in a language different from the language of the source documents. In this paper, we propose a graph-based approach to multi-document summarization that integrates machine translation quality scores in the sentence extraction process. We evaluate our method on a manually translated subset of the DUC 2004 evaluation campaign. Results indicate that our approach improves the readability of the generated summaries without degrading their informativity.**

***Index Terms*—Graph-based approach, cross-language multi-document summarization.**

I. Introduction

THE rapid growth and online availability of information in numerous languages have made cross-language information retrieval and extraction tasks a highly relevant field of research. Cross-language document summarization aims at providing a quick access to information expressed in one or more languages. More precisely, this task consists in producing a summary in one language different from the language of the source documents. In this study, we focus on English to French multi-document summarization. The primary motivation is to allow French readers to access the ever increasing amount of news available through English news sources.

Recent years have shown an increased amount of interest in applying graph theoretic models to Natural Language Processing (NLP) [1]. Graphs are natural ways to encode information for NLP. Entities can be naturally represented as nodes and relations between them can be represented as edges. Graph-based representations of linguistic units as diverse as words, sentences and documents give rise to efficient solutions in a variety of tasks ranging from part-of-speech tagging to information extraction, and sentiment analysis. Here, we apply a graph-based ranking algorithm to multi-document summarization.

A straightforward idea for cross-language summarization is to translate the summary from one language to the other. However, this approach does not work well because of the errors committed by Machine Translation (MT) systems. Indeed, translated sentences can be disfluent or difficult to understand. Instead, we propose to consider the translation quality of the French sentences in the sentence selection process. More precisely, we use a supervised learning approach to predict MT quality scores and integrate these scores during the graph construction.

This paper is organized as follows. We first briefly review the previous work, followed by a description of the method we propose. Next, we present our experiments and results. Lastly, we conclude with a discussion and directions for further work.

Manuscript received November 9, 2010. Manuscript accepted for publication January 15, 2011.

Florian Boudin and Stéphane Huet are with Université d'Avignon, France. Juan-Manuel Torres-Moreno is with Université d'Avignon, France; École Polytechnique de Montréal, Canada; Universidad Nacional Autónoma de México, Mexico (e-mail: firstname.lastname@univ-avignon.fr).

II. Related Work

A. Predicting Machine Translation Quality

Machine translation is a natural component for cross-language document summarization. However, as an automatic process, MT systems are prone to generate errors and thus to mislead summarization. These errors can either introduce wrong information with respect to the source-language documents to summarize or make sentences disfluent and difficult to understand. In order to alleviate these effects, it is relevant to take into account a score that assesses the translation quality and that can be used to filter out incorrect translations during summarization.

Predicting quality translation, referred to as confidence estimation in the MT domain, has first been viewed as a binary classification problem to distinguish good translations from bad ones [4]. More recent studies have been done to estimate a continuous quality score at the word level [19] or at the sentence level [19], [20]. In this paper, we choose to resort to sentence-level quality scores that are more easily integrated into the summarization sentence extraction process.

Various classifiers have been used to estimate translation quality. Statistic models are trained on a set of translations manually labeled as correct or incorrect [17], [20] or tagged through automatic metrics like word error rate [4], NIST [4], [20] or BLEU scores [19]. Various features are extracted to compute quality values: linguistic features depending or not on resources like parsers or Wordnet, similarity features between the source sentence and the target sentence and some internal features of the MT system, such as the alternative

translation per source words or the phrase scores of n-best list of translation candidates.

B. Graph-Based Summarization

Extensive experiments on multi-document summarization have been carried out over the past few years, especially through the DUC (Document Understanding Conference) evaluations.[1] Most of the proposed approaches are based on an extraction method, which identifies salient textual segments, most often sentences, in documents. Sentences containing the most salient concepts are selected, ordered and assembled according to their relevance to generate summaries (also called extracts).

Previous work on multi-document summarization includes, among others, centroid-based sentence selection [18], supervised learning [22], and information fusion [2]. The interested reader is directed to the DUC proceedings for more information on the various approaches. In this paper, we concentrate on graph-based ranking approaches. The rest of this section presents the previous work relevant to this type of summarization.

Approximately at the same time, Erkan and Radev [9] and Mihalcea [13] proposed to apply graph-based ranking algorithms to sentence extraction. The underlying idea is that of representing documents as graphs. Sentences are represented as nodes and relations between them, e.g. similarity measures, are represented as edges. Ranking algorithms are a way of deciding on the importance of a node, i.e. a sentence, based on the information drawn from the entire graph. Such approaches have several advantages. First, differently from most other methods, they do not require training data. Second, they are easily adaptable to other languages [14].

C. Cross-language Summarization

Cross-language summarization has received much attention recently and several approaches have been proposed. A natural way to go about this task would be to translate the documents prior to summarization, or to translate the generated summary. Orăsan and Chiorean [15] proposed to use the Maximal Marginal Relevance (MMR) method [6] to produce Romanian news summaries and then automatically translate them into English. More recently, Wan *et al.* [21] showed that incorporating translation quality scores in the summarization process increases both generated summary' content and readability. They focused on English-to-Chinese mono-document summarization and employed supervised learning to predict MT quality. In this study we will go a step further by incorporating MT confidence scores in cross-language multi-document summarization. Unlike the work of Wan *et al.*, our approach uses an unsupervised language-independent ranking algorithm for sentence selection [14].

[1]Document Understanding Conferences were conducted from 2000 to 2007 by the National Institute of Standards and Technology (NIST), `http://duc.nist.gov`

III. Method

In this section, we describe our method for cross-language multi-document summarization. We based our approach on a two-step summarization process which first scores each sentence, and then selects the top ranked sentences for inclusion in the summary. A preliminary step is added in order to translate each sentence and estimate the resulting translation quality. We modified the graph construction step to take advantage of the translation quality scores. Lastly, the French summary is constructed from the translation of the top ranked English sentences. Figure 1 presents an overview of the architecture of our proposed method.

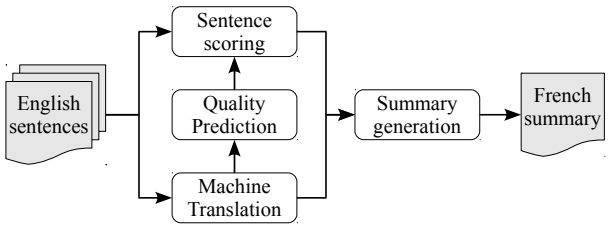


Fig. 1. Architecture of our proposed summarization system.

A. Pre-processing Documents and MT Quality Prediction

Each document in the cluster is segmented into sentences using the Punkt sentence boundary detection method [11] implemented in the NLTK toolkit [3]. All the English sentences were automatically translated into French using the Google translate service.[2]

An MT score is computed for each sentence to estimate both the translation accuracy and the fluency of the generated French sentences. This score aims at promoting in the summarization process sentences that can be easily read and understood by French speaking readers. In order to obtain it, we computed for each sentence 8 features that provide information on how difficult the source sentence is and how fluent the generated translation is:

- the source language sentence length in terms of words,
- the ratio of source and target lengths,
- the number of punctuation marks in the source language sentence,
- the proportion of the source numbers and punctuation symbols found in the target sentence,
- the perplexities of the source and the target sentences computed by 5-gram forward Language Models (LMs),
- the perplexities of the source and the target sentences computed by 2-gram backward LMs, i.e after reversing the word order of sentences.

[2]`http://translate.google.com`

These first four features belong to the most relevant features underlined by [20], among 84 features studied; the last four ones have already turned out to be effective for sentence-level confidence measures [19]. LMs are built using monolingual corpora of the news domain, made available for the WMT 10 workshop [5] and consisting of 991M English words and 325M French words. Perplexity scores are expected to reflect fluency, the use of 2-gram backward LMs addressing more specifically the detection of incorrect determinants or other function words. Contrary to other studies, we decided to focus on basic features that does not require any linguistic resources, such as parsers or dictionaries. Besides, features were restrained to scores computed only from the input sentence and its translated sentence, and therefore do not depend on the MT system used.

To predict MT quality from features, we adopt the ϵ-Support Vector Regression method (ϵ-SVR), already used for this purpose [21], [19]. In our experiments, we resort to the LIBSVM library [7] using the radial basis function as kernel, as recommended by the authors. The regression model depends on two parameters: an error cost c and a coefficient γ of the kernel function; their values have been optimized on a training corpus by grid search and cross-validation.

Ideally, the ϵ-SVR model should be trained on a corpus labeled with human judgments of MT output quality. Unfortunately, we are not aware of a large enough corpus of this kind for the English-French pair and producing MT judgments is a very slow process. We decided to resort instead to the automatic metric NIST [8] as an indicator of quality. Indeed, this metric have already been used in the past for this purpose [4], [20] and turned out to be more correlated with human judgments at the sentence level than other metrics such as the widely used BLEU [4]. Our training corpus was built from the reference translations provided in the news domain for the WMT workshops [5] from 2008 to 2010, which represents a set of 7,112 sentences. In order to assess the quality of the so-built model, we computed the Mean Squared Error (MSE) metric: $\frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$, where N is the number of sentences, $\hat{y}$ is the prediction estimated by the regressor and y the actual value. On the 2,007 sentences made available for WMT 07 and kept for this purpose, we obtained a MSE of 0.456.

B. Sentence Scoring

We use a graph-based ranking approach to multi-document summarization. The first step is to construct a graph that represents the text. Let $G = (V, E)$ be a directed graph with the set of vertices (nodes) V and a set of directed edges E, where E is a subset of $V \times V$. Let $pred(V_i)$ be the set of vertices that point to the vertex V_i and $succ(V_i)$ the set of vertices that vertex V_i points to. A node is added to the graph for each sentence in the cluster. Connections (edges) between sentences (nodes) are defined in terms of similarity. We use the similarity measure proposed in [13], computed as a function of content overlap. The overlap of two sentences is the number of common tokens between the lexical representations of the two sentences, after stop words removal and stemming with the Porter stemmer. To avoid promoting long sentences, this number is normalized by the sentence lengths. Given $\mathrm{freq}(w, S)$ the frequency of word w in sentence S, the similarity between S_i and S_j is defined as:

$$Sim(S_i, S_j) = \frac{\sum_{w \in S_i, S_j} \mathrm{freq}(w, S_i) + \mathrm{freq}(w, S_j)}{\log(|S_i|) + \log(|S_j|)} \quad (1)$$

Graph-based ranking algorithms implements the concept of recommendation. Sentences are scored by taking into account global information recursively computed from the entire graph. In this study, we use an adaptation of the Google's PageRank ranking algorithm [16] to include edge weights:

$$p(V_i) = (1-d) + d \times \sum_{V_j \in pred(V_i)} \frac{Sim(S_i, S_j)}{\sum_{V_k \in succ(V_i)} Sim(S_k, S_i)} p(V_i) \quad (2)$$

where d is a "damping factor", which is typically chosen in the interval $[0.8, 0.9]$ (see [16]). This method, described in [13], is very similar to Lexical PageRank (LexRank) [9]. From a mathematical point of view, the PageRank algorithm computes the dominant eigenvector of the matrix representing the graph. We will use this method as baseline in our experiments.

C. Incorporating MT Quality Scores

In order to address the cross-language aspect, machine translation quality scores are introduced at the graph construction step. We modified Equation 1 to:

$$Sim_2(S_i, S_j) = Sim(S_i, S_j) \times Prediction(S_i) \quad (3)$$

where $Prediction(S_i)$ is the translation quality score of sentence S_i computed in Section III-A. Unlike the similarity measure defined by Equation 1 which is symmetric, this measure is directed. An accurate and fluent translated sentence would have its outgoing edge weights strengthen and hence would play a more central role in the graph. This way, sentences that are both informative and that are predicted to be accurately translated by the MT system will be selected.

We made some adaptations to the ranking algorithm to take advantage of the specificity of the documents. The position of a sentence within a document is a strong indicator of the importance of its content. This is especially true in newswire articles, which tend to always begin with a concise description of the subject of the article. Thus, double weight is given to all edges outgoing from a node corresponding to a leading sentence. Lastly, identical sentences (we keep only one occurrence) and sentences less than 5 word long are automatically dismissed.

D. Summary Generation

It is often the case that clusters of multiple documents, all related to the same topic, contain very similar or even identical sentences. To avoid such pairs of sentences, which may decrease both readability and content aspects of the summary, we have to use a redundancy removal method. Maximal Marginal Relevance (MMR) [6] is perhaps the most widely used redundancy removal technique. It consists in iteratively selecting summary sentences that are both informative and different from the already selected ones. In her work, Mihalcea introduces a maximum threshold on the sentence similarity measure [14]. Accordingly, at the graph construction step, no edge is added between nodes (sentences) whose similarity exceeds this threshold. In this study, we choose to use a two-step sentence selection method for maximizing the amount of information conveyed in the summary and minimizing the redundancy.

The second sentence selection step determines among the top scored sentences, as evaluated in the sentence ranking step, those which would make the best summary when combined together [10]. We first generate all the candidate summaries from combinations of the N sentences with the best relevance score that have the following properties: their combined number of characters does not exceed a threshold $\mathcal{T}$; no other sentences can be added while still remaining under a number of characters $\mathcal{T}$. Each candidate summary is then scored using a combination of word diversity (number of unique n-grams for $n \in [1, 2]$) and sentence relevance (sum of individual sentence scores). The sentences contained in the candidate summary with the best global score are the ones selected for the summary.

Summaries are constructed by sorting the selected sentences in chronological order to maximize temporal coherence. Sentences extracted from the oldest documents are displayed first. If two sentences are extracted from the same document, the original order within the document is kept.

IV. Results

In this section, we describe the details of our experimental protocol. We first give a description of the data set and the evaluation metrics we used. Then, we present the results obtained by our cross-language summarization system.

A. Experimental Settings

In this study, we used the document sets made available during the Document Understanding Conference (DUC) 2004 evaluation. DUC 2004 provided 50 English document clusters for generic multi-document summarization. Each cluster contains on average 10 newswire documents from the Associated Press and New York Times newswires. The task consists in generating short summaries representing all the content of the document set to some degree. Summaries must not exceeds 665 characters (alphanumerics, white spaces and punctuation included). This maximum length was derived from the manual summaries used in DUC 2003. We performed both automatic evaluation of content and manual evaluation of readability on a subset of the DUC 2004 data set made of 16 randomly selected clusters.

1) Automatic Evaluation: The majority of existing automated evaluation methods work by comparing the generated summaries to one or more reference summaries (ideally, produced by humans). To evaluate the quality of our generated summaries, we choose to use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [12] evaluation toolkit, that has been found to be highly correlated with human judgments. ROUGE is a n-gram recall-based measure calculated as the number of overlapping n-grams between a candidate summary and a set of reference summaries. In our experiments, three metrics are computed: ROUGE-1 (unigram-based), ROUGE-2 (bigram-based) and ROUGE-SU4 (skip-bigram, allowing bigrams to be composed of non-contiguous words with as many as four words intervening). We run the version 1.5.5 of ROUGE with the default parameters[3] given by the DUC guidelines.

Reference English summaries for DUC 2004 were provided by NIST annotators. Four reference summaries were manually produced for each cluster. In our work, we focused on generating French summaries from English document sets. To be able to evaluate our method, we asked three annotators to translate the subset of 16 cluster's English reference summaries into French reference summaries. The translation instructions the annotators were given are fairly simple: each summary is to be translated sentence by sentence without introducing any kind of extraneous information (e.g. anaphora generation, proper name disambiguation or any sentence reduction technique). 64 reference summaries were translated this way, four for each cluster. The translators spent on average 15 minutes per summary (a total of more than 16 hours).

We have not restricted the size of the translated summaries to a given length. Accordingly, the length of the French reference summaries is on average 25% longer (in number of characters) than English ones. Similarly, our generation algorithm does not impose a maximum length on the French summaries but uses the total length of the corresponding English sentences. Lastly, we adapted the Porter stemmer embedded in the ROUGE evaluation package to correctly handle French words.

2) Manual Evaluation: The linguistic well-formedness of each summary is evaluated using a protocol similar to the one used during the DUC campaigns. We evaluate the readability aspect of the summaries on a five-point scale from 1 to 5, where 5 indicates that the summary is "easy to read", and 1 indicates that the summary is "hard to read". Annotators were asked to grade two randomly ordered summaries, one generated with the proposed method and the other obtained by translating the English output of a state-of-the-art approach

[3] `ROUGE-1.5.5.pl -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d`

(described in Section III-B). Five annotators participated in the manual evaluation.

B. Monolingual Experiments

We first wanted to investigate the performance of the described method on a monolingual summarization task. Table I reports the automatic evaluation scores obtained on the DUC 2004 data set for different sentence scoring methods. *Graph-Sum* stands for the graph-based ranking method presented in Section III-B. Baseline results are obtained on summaries generated by taking the leading sentences of the most recent documents of the cluster, up to 665 characters (official baseline of DUC, identifier is 2). The table also lists the top performing system (DUC identifier is 65) at DUC 2004. We observe that the graph-based ranking approach achieves state-of-the-art performance, the difference with the best system is not statistically significant (paired Student's t-test of $\rho = 0.77$ for ROUGE-1, $\rho = 0.17$ for ROUGE-2 and $\rho = 0.57$ for ROUGE-SU4). By ways of comparison our system would have been ranked in the top 4 at the DUC 2004 campaign. Moreover, no post-processing was applied to the selected sentences leaving an important margin of progress.

TABLE I
ROUGE AVERAGE RECALL SCORES COMPUTED ON THE DUC 2004 DATA SET, THE RANK AMONG THE 35 PARTICIPANTS IS ALSO GIVEN. SCORES MARKED WITH † ARE STATISTICALLY SIGNIFICANT OVER THE BASELINE (PAIRED STUDENT'S T-TEST WITH $\rho < 0.001$)

System	ROUGE-1	**rank**	ROUGE-2	**rank**	ROUGE-SU4	**rank**
1^{st} system	0.38244†	1	0.09218†	1	0.13323†	1
Graph-Sum	0.38052†	2	0.08566†	4	0.13114†	3
Baseline	0.32381	26	0.06406	25	0.10291	29

C. Cross-language Experiments

In this second series of experiments, we evaluated our method for cross-language multi-document summarization. Baseline results are obtained by translating the English output of the graph-based ranking approach (described in Section III-B). The automatic ROUGE evaluation scores are presented in Table II. We observe a small improvement in ROUGE-2 and ROUGE-SU4 for our method. Nevertheless, this increase is not significant. This result can be explained by the fact that MT quality scores can promote inside the summary some sentences that are less informative but more understandable and readable.

TABLE II
ROUGE AVERAGE RECALL SCORES COMPUTED ON THE FRENCH TRANSLATED SUBSET OF THE DUC 2004 DATA SET

System	ROUGE-1	ROUGE-2	ROUGE-SU4
Baseline	0.39704	0.10249	0.13711
Our method	0.39624	0.10687	0.13877

We then evaluated the linguistic well-formedness of the summaries generated with our proposed method. Table III shows the manual evaluation results on the subset of 16 clusters. The average score given by each human judge is also given. We observe that the proposed approach obtains better readability scores. All annotators agree that our method produces more easy-to-read summaries than the baseline. This result indicates that MT quality scores are useful for selecting more readable sentences. An example of generated summaries is given in Appendix 1. Overall, results show that our method can enhance the readability of the generated summaries without degrading their informativity. However, the average readability scores are relatively low. An analysis of the errors observed in French summaries leads us to think that pre-processing source sentences (e.g. removing ungrammatical sentences) can be a first step to filter out erroneous sentences.

TABLE III
READABILITY SCORES OF OUR PROPOSED METHOD COMPARED TO THE STANDARD GRAPH-BASE RANKING APPROACH (BASELINE). SCORES ARE ON A FIVE-POINT SCALE FROM 1 TO 5, WHERE 5 INDICATES THAT THE SUMMARY IS "EASY TO READ", AND 1 IS "HARD TO READ"

Annotator	**Readability**	
	Baseline	Our method
Annotator 1	2.44	2.50
Annotator 2	1.56	1.63
Annotator 3	1.75	2.31
Annotator 4	3.06	3.31
Annotator 5	1.50	1.63
Average	2.06	2.28

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a graph-based approach to cross-language multi-document summarization. We proposed to introduce machine translation quality scores at the graph construction step. Automatically translated sentences that are both fluent and informative are then selected by our ranking algorithm. We evaluated our approach on a manually translated subset of 16 clusters from the DUC 2004 data set. Results show that our approach enhances the readability of the generated summaries without degrading their content.

In future work, we intend to expand the set of reference summaries by translating the entire DUC 2004 data set. We also plan to extend the evaluation to other languages. The manually translated French summaries introduced in this paper, along with the manual given to the group of translators, is available for download on request.

REFERENCES

[1] C. Banea, A. Moschitti, S. Somasundaran, and F. M. Zanzotto, Eds., *Proceedings of TextGraphs-5 Workshop*, Uppsala University. Uppsala, Sweden: ACL, 2010. [Online]. Available: http://www.aclweb.org/anthology/W10-23

[2] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," Johns Hopkins University, Batimore, MD, USA, Tech. Rep., 2003.

[3] S. Raybaud, D. Langlois, and K. Smaïli, "Efficient combination of confidence measures for machine translation," in *Proceedings of Interspeech 2009 conference*, Brighton, UK, 2009, pp. 424–427.
[4] L. Specia, N. Cancedda, M. Dymetman, M. Turchi, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *Proceedings of EAMT 2009 conference*, Barcelona, Spain, 2009, pp. 28–35.
[5] C. B. Quirk, "Training a sentence-level machine translation confidence measure," in *Proceedings of LREC 2004 conference*, Lisbon, Portugal, 2004, pp. 825–828.
[6] D. Radev, H. Jing, M. Sty, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919–938, 2004.
[7] K.-F. Wong, M. Wu, and W. Li, "Extractive summarization using supervised and semi-supervised learning," in *Proceedings of Coling 2008 conference*, Manchester, UK, 2008, pp. 985–992. [Online]. Available: http://www.aclweb.org/anthology/C08-1124
[8] R. Barzilay, K. R. McKeown, and M. Elhadad, "Information fusion in the context of multi-document summarization," in *Proceedings of ACL 1999 conference*, College Park, MD, USA, 1999, pp. 550–557. [Online]. Available: http://www.aclweb.org/anthology/P99-1071
[9] G. Erkan and D. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *JAIR*, vol. 22, no. 1, pp. 457–479, 2004.
[10] R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of ACL 2004 conference*, Barcelona, Spain, July 2004, pp. 170–173.
[11] R. Mihalcea and P. Tarau, "A language independent algorithm for single and multiple document summarization," in *Proceedings of IJCNLP 2005 conference*, vol. 5, Jeju Island, South Korea, 2005.
[12] C. Orăsan and O. A. Chiorean, "Evaluation of a cross-lingual romanian-english multi-document summariser," in *Proceedings of LREC 2008 conference*, Marrakech, Morocco, 2008. [Online]. Available: http://clg.wlv.ac.uk/papers/539_paper.pdf
[13] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of SIGIR 1998 conference*. ACM, 1998, pp. 335–336.
[14] X. Wan, H. Li, and J. Xiao, "Cross-language document summarization based on machine translation quality prediction," in *Proceedings of ACL 2010 conference*, Uppsala, Sweden, 2010, pp. 917–926. [Online]. Available: http://www.aclweb.org/anthology/P10-1094
[15] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006.
[16] S. Bird and E. Loper, "Nltk: The natural language toolkit," in *Proceedings of ACL 2004 conference*, Barcelona, Spain, 2004, pp. 214–217.
[17] C. Callison-Burch, P. Koehn, C. Monz, K. Peterson, M. Przybocki, and O. Zaidan, "Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation," in *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR (WMT)*, Uppsala, Sweden, 2010, pp. 17–53. [Online]. Available: http://www.aclweb.org/anthology/W10-1703
[18] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
[19] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of HLT 2002 conference*, San Diego, CA, USA, 2002, pp. 138–145.
[20] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford Digital Library Technologies Project, Tech. Rep., 1998.
[21] P. Genest, G. Lapalme, L. Nerima, and E. Wehrli, "A symbolic summarizer with 2 steps of sentence selection for tac 2009," in *Proceedings of TAC 2009 Workshop*, Gaithersburg, MD, USA, 2009.
[22] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of ACL-04 Workshop*, S. S. Marie-Francine Moens, Ed., Barcelona, Spain, 2004, pp. 74–81.

APPENDIX 1

TABLE IV
EXAMPLE OF FRENCH SUMMARIES GENERATED FOR THE DUC CLUSTER D30007T BY THE BASELINE AND THE PROPOSED APPROACH

Baseline (average readability score of 2.4)

Après une journée de combats, les rebelles congolais a annoncé dimanche avoir conclu Kindu, la ville stratégique et à la base dans l'est du Congo utilisé par le gouvernement pour mettre fin à leurs avances. (*After a day of fighting, Congolese rebels said Sunday they had entered Kindu, the strategic town and airbase in eastern Congo used by the government to halt their advances.*) Etienne Ngangura, un porte-parole des rebelles, a déclaré les combattants rebelles se trouvaient dans Kindu et avait pris le côté, grande base aérienne, 380 km (235 miles) à l'ouest de Goma, le fief des rebelles. (*Etienne Ngangura, a rebel spokesman, said the rebel fighters were inside Kindu and had taken the adjacent, large airbase, 380 kilometers (235 miles) west of Goma, the rebel stronghold.*) "Nos soldats sont dans la ville et les combats se poursuivent", le commandant de bataillon rebelle Arthur Mulunda a déclaré à Kalima, à 80 kilomètres (50 miles) au nord de Kindu. (*"Our soldiers are in the town and the fighting is continuing" rebel battalion commander Arthur Mulunda said in Kalima, 80 kilometers (50 miles) northeast of Kindu*) Le samedi, les rebelles ont dit qu'ils ont abattu un Boeing 727 Congolais qui tentait d'atterrir à la base aérienne de Kindu avec 40 troupes et de munitions. (*On Saturday, the rebels said they shot down a Congolese Boeing 727 which was attempting to land at Kindu air base with 40 troops and ammunition*)

Our method (average readability score of 3.2)

Les rebelles ont attaqué un village dans l'ouest de l'Ouganda et a tué six civils devant des soldats contraints de rebrousser chemin, une porte-parole militaire a déclaré jeudi. (*Rebels attacked a village in western Uganda and killed six civilians before soldiers drove them off, a military spokesman said Thursday*) Etienne Ngangura, un porte-parole des rebelles, a déclaré les combattants rebelles se trouvaient dans Kindu et avait pris le côté, grande base aérienne, 380 km (235 miles) à l'ouest de Goma, le fief des rebelles. (*Etienne Ngangura, a rebel spokesman, said the rebel fighters were inside Kindu and had taken the adjacent, large airbase, 380 kilometers (235 miles) west of Goma, the rebel stronghold*) Les commandants rebelles, a déclaré mardi qu'ils étaient sur le point d'envahir une importante base aérienne détenue par le gouvernement au Congo Est, une bataille qui pourrait déterminer le futur de la guerre de deux mois congolais. (*Rebel commanders said Tuesday they were poised to overrun an important government-held air base in eastern Congo, a battle that could determine the future of the two-month Congolese war*) Les rebelles dans l'est du Congo a déclaré samedi qu'ils ont abattu un avion de ligne transportant 40 soldats du gouvernement dans un aéroport stratégique face à un assaut des rebelles. (*Rebels in eastern Congo on Saturday said they shot down a passenger jet ferrying 40 government soldiers into a strategic airport facing a rebel assault*)

Journal Information and Instructions for Authors

I. Journal Information

"*Polibits*" is a half-yearly research journal published since 1989 by the Center for Technological Design and Development in Computer Science (CIDETEC) of the Instituto Politécnico Nacional ((IPN) in Mexico City, Mexico. The journal solicits original research papers in all areas of computer science and computer engineering, with emphasis on applied research.

The journal has double-blind review procedure. It publishes papers in English and Spanish.

Publication has no cost for the authors.

A. Main Topics of Interest

The journal publishes research papers in all areas of computer science and computer engineering, with emphasis on applied research.

More specifically, the main topics of interest include, though are not limited to, the following:

- Artificial Intelligence
- Natural Language Processing
- Fuzzy Logic
- Computer Vision
- Multiagent Systems
- Bioinformatics
- Neural Networks
- Evolutionary algorithms
- Knowledge Representation
- Expert Systems
- Intelligent Interfaces: Multimedia, Virtual Reality
- Machine Learning
- Pattern Recognition
- Intelligent Tutoring Systems
- Semantic Web
- Database Systems
- Data Mining
- Software Engineering
- Web Design
- Compilers
- Formal Languages
- Operating Systems
- Distributed Systems
- Parallelism
- Real Time Systems
- Algorithm Theory
- Scientific Computing
- High-Performance Computing
- Geo-processing
- Networks and Connectivity
- Cryptography
- Informatics Security
- Digital Systems Design
- Digital Signal Processing
- Control Systems
- Robotics
- Virtual Instrumentation
- Computer Architecture
- other.

B. Indexing

LatIndex, Periódica, e-revistas, index of excellence of CONACYT (Mexico).

II. Instructions for Authors

A. Submission

Papers ready to review are received through the Web submission system www.easychair.org/polibits. See also the updated information at the web page of the journal www.cidetec.ipn.mx/polibits.

The papers can be written in English or Spanish.

Since the review procedure is double-blind, the full text of the papers should be submitted without names and affiliations of the authors and without any other data that reveals the authors' identity.

For review, a file in one of the following formats is to be submitted: PDF (preferred), PS, Word. In case of acceptance, you will need to upload your source file in Word or TeX. We will send you further instructions on uploading your camera-ready source files upon acceptance notification.

Deadline for the nearest issue (July-December 2011): October 1, 2011. Papers received after this date will be considered for the next issues.

B. Format

Please, use IEEE format[1], see section "Template for all Transactions (except IEEE Transactions on Magnetics)". The editors keep the right to modify the format and style of the final version of the paper if necessary.

Please, follow carefully instructions for formatting of the references. If you use TeX, you should include the bib file.

We do not have any specific page limit: we welcome both short and long papers, provided the quality and novelty of the paper adequately justifies the length.

In case of being written in Spanish, the paper should also contain the title, abstract, and keywords in English.

[1] www.ieee.org/web/publications/authors/transjnl/index.html