

Revised N-Gram based Automatic Spelling Correction Tool to Improve Retrieval Effectiveness

Farag Ahmed, Ernesto William De Luca, and Andreas Nürnberger

Abstract—We present a language-independent spell-checker that is based on an enhancement of the n-gram model. The spell checker is proposing correction suggestions by selecting the most promising candidates from a ranked list of correction candidates that is derived based on n-gram statistics and lexical resources. Besides motivating and describing the developed techniques, we briefly discuss the use of the proposed approach in an application for keyword- and semantic-based search support. In addition, the proposed tool was compared with state-of-the-art spelling correction approaches. The evaluation showed that it outperforms the other methods.

Index terms—Spelling correction, n-gram, information retrieval effectiveness.

I. INTRODUCTION

THE problem of devising algorithms and techniques to automatically correct words in texts has become a perennial research challenge. Work began as early as the 1960s on computer techniques for automatic spelling correction and automatic text recognition, and it has continued up to the present. There are good reasons for the continuing research efforts in this area in order to improve quality and performance and to broaden the spectrum of possible applications [1]. For example, even though system programs (language processors, operating systems, etc.) have become increasingly powerful and sophisticated, they do not assist the user (with very few exceptions) in correcting many of the obvious spelling errors in the source input. There are two types of word errors, the real-word error and the non-word error. Real-word errors are misspelled words that have a meaning and can be found in a dictionary. Non-word errors are words that have no meaning and are thus not included in a dictionary. We concentrate on the correction of the non-word error with the proposed algorithm. Damerau (1964) found that 80% of misspelled words that are non-word errors are the result of a single insertion, deletion, substitution or transposition of letters [2]. Therefore, it seems reasonable to base correction algorithms on measures that consider these simple operations. However, approaches based on pure n-

gram statistics (which account for these operations implicitly) have also proven to provide good performance [1, 15].

In this paper, we propose an approach that is based on an enhancement of the n-gram model. Therefore, we first discuss briefly, related work on spelling correction in Section 2. Afterwards, we describe, in detail, in Section 3 our spell checking approach MultiSpell. In Section 4, we present an evaluation based on benchmark data sets in the English and Portuguese language and conclude with a brief discussion.

II. APPROACHES OF SOME SPELL CHECKERS

Algorithmic techniques for detecting and correcting spelling errors in text have a long and robust history in computer science [1]. Many approaches have been applied since people started to deal with this problem. Different techniques like edit distance [4], rule-based techniques [10], n-grams [20], probabilistic techniques [14], neural nets [15], similarity key techniques [16, 17] and noisy channel model [18, 19] have been proposed. All of these are based on the idea of calculating the similarity between the misspelled word and the words contained in a dictionary. In the following, we describe briefly one of the most popular approaches (Aspell) and one recently proposed approach for the Portuguese language (TST) [13] that we used for comparison.

GNU Aspell, usually called just Aspell, is a standard spell-check software for the GNU software system. There are dictionaries for about 70 languages available. GNU Aspell is a Free and Open Source and can be downloaded under <http://aspell.sourceforge.net/>. In contrast to Ispell, which suggests words with small edit-distance, Aspell in addition compares sounds-like equivalents (computed for English words using the metaphone algorithm [21]) up to a given edit distance.

The Ternary Search Trees [13] approach (TST) is a dictionary data structure working with string-keys. It can find, remove and add these keys quickly and also easily search the tree for partial matches. Additionally, near-match functions can be implemented. These give the possibility to suggest alternatives for misspelled words.

For a more conclusive overview of spell-check approaches see [1, 15].

Manuscript received October 23, 2008. Manuscript accepted for publication August 22, 2009.

Farag Ahmed and Andreas Nürnberger are with Data and Knowledge Engineering Group, Institute for Knowledge and Language Engineering, Otto-von-Guericke University of Magdeburg, Germany.

Ernesto William De Luca is with Competence Center Information Retrieval & Machine Learning Distributed Artificial Intelligence Laboratory, Technical University of Berlin, Germany.

$w_{4,5}$ and $w_{5,6}$ of the correction candidate w , i.e. even if the n-gram $w'_{4,5}$ is similar to $w_{2,3}$ this would not count towards the similarity score of the words w' and w .

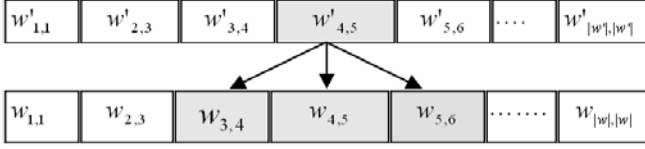


Fig. 1. Bigram comparison for misspelled word w' and a correction candidate w using a comparison window of size 3. Notice that the first and last n-gram represent the first and the last letters only and are therefore always of size one.

Overall, the computation of the similarity score S for a given n-gram size n and a given odd-numbered window size m can be defined as follows, assuming that u is the longer word (if v is longer than u and v can simply be exchanged):

$$S_{n,m}(u,v) =$$

$$\frac{g(u_{1,1}, v_{1,1}) + g(u_{|u|,|u|}, v_{|v|,|v|}) + \sum_{i=2}^{|u|-n+1} \sum_{j=\frac{m-1}{2}}^{\frac{m-1}{2}} g(u_{i,i+(n-1)}, v_{i+j,i+j+(n-1)})}{N} \quad (2)$$

$$\text{where } g(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \quad \text{and}$$

$$u_{i,j} = \begin{cases} \text{substring}(u, i, j) & \text{if } i \leq j \\ "" & \text{otherwise.} \end{cases}$$

Here, $g(u_{1,1}, v_{1,1})$ compares the first and $g(u_{|u|,|u|}, v_{|v|,|v|})$ the last characters of the words u and v and the nested sum counts the number of n-grams in v that are similar to n-grams in a window of size m around the same position in word v . N is computed similarly as in Eq. (1). In Fig. 2 the specific cases that have to be considered when computing the similarity score S are summarized.

D. The MultiSpell Algorithm

The first stage of the MultiSpell algorithm is to compare the keywords given from the user with the correct words contained in the dictionary. First of all, we check based on the used dictionary (here, based on the words extracted from MultiWordNet) if the word is misspelled. If this is the case, the algorithm builds n-grams for the misspelled word. Then we select correction candidates from the dictionary. In order to keep the number of correction candidates as small as possible, we select only words as candidates that are two characters shorter or longer than the misspelled word. This is motivated by the work of Turba [11], who has shown that most misspelled words differ in length only by one character from the correct word.

For the selected words the n-grams are computed and the similarity score is computed according to Eq. (2). The correction candidates can then be simply sorted by the obtained similarity score and the word with the highest score is proposed as the best correction candidate.

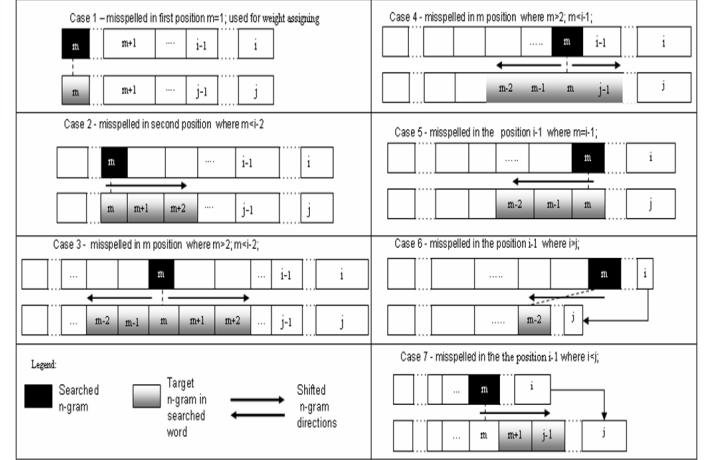


Fig. 2. Comparing n-grams based on the MultiSpell algorithm.

E. Spelling Correction for Keyword- and Semantic-based Search Support

MultiSpell has been also integrated as a pre-processing approach in the Sense Folder Framework [25]. It can be applied to queries and documents, in order to support users during keyword-based and semantic-based search. The first is an important task for retrieving the relevant documents related to the query identifying the misspelled words and correct them for a correct interpretation [23] (see also Fig. 3). The second is specifically trying to improve the semantic search process [24]; therefore several problems have to be addressed, before the semantic classification of documents is started. When users mistype the query in writing, the system has to be able to give correction alternatives to continue the semantic-based search.

The semantic-based search differs from the “normal” search, because users are “redirected” to semantic concepts that could describe their query. This semantic support is provided in the user interface. On the left side of the user interface (see Fig. 4) suggestions are generated by MultiSpell and presented to the user for starting the semantic-based search.

In this case, the use of MultiSpell is mostly helpful, not only because it performs an efficient correction (as shown in Fig. 3), but also because it can “redirect” the user to a semantic search (see Fig. 4). Thus, if the user types a word that is not contained in the lexical resource used, the system can suggest other “similar” words according to the words found in the resource. Then, a semantic classification is started using the words selected by the user.

IV. COMPARISON AND EVALUATION OF RESULTS

In the following, we show results of some experiments done for the English and Portuguese language. The first evaluation was done on the whole English commonly misspelled word list provided in [12]. Afterwards, we compared the results of our spell checker MultiSpell with the results of the TST approach (in one experiment, for the Portuguese language) and of the Aspell approach (in two experiments, for the Portuguese and the English language), showing that the proposed approach always achieved the best results.

For the first evaluation, we used the whole list of commonly misspelled words in English consisting of 3975 words as published in [12]. This list of common spelling mistakes is represented by a table consisting of two columns. The first one shows the misspelled word, the second the correct spelling. For the evaluations, we only considered the correction words that were ranked as best correction word, i.e., even if the second word would have been the correct candidate, this was counted as a wrong correction. We first used all misspelled words of the list, using the bigram case and just the first candidate correction. MultiSpell corrected 3334 misspelled words (84%) and failed for 641 misspelled words (16%) although it provided similar corrections in many cases. For example the word *advice* was suggested instead of *advised* for the misspelled word *advised*. Another example is the provided correction *algebraically* instead of *algebraic* for the misspelled word *algebraical* (see Table V in the Appendix). These suggestions were classified as wrong in our approach, even though they belong to the same word sense. Second, we used trigrams. This showed lower performance and efficiency. MultiSpell corrected 2900 words (73%) and failed for 1075 (27%) as shown in Table II.

A. Evaluation of English Spelling Correction

For the second evaluation, we randomly selected a set of only 120 misspelled words obtained from Wikipedia [12] and not the whole list. All error types and starting letters of the words were taken into account. We compared MultiSpell with Aspell, MicrosoftWord, and Google. Since Aspell provides a list of candidate corrections we took just the first candidate from the list assuming that the first candidate is the most likely one proposed by the algorithm. MicrosoftWord and Google provided only one correction candidate. Table III and Table V (in the Appendix) show that MultiSpell finds the correct spelling for 109 words (90%). In comparison, Google can correct 106 (88%) words, while Aspell and MicrosoftWord 105 words (87.5%). MultiSpell detected 6 of 16 of the multiple correction words (which have more than one possible correction), but it doesn't fail to provide at least one correct suggestion. Aspell detected just two of the multiple corrections and it failed just one time to provide a suggestion for one of the multiple corrections.

TABLE II
COMPARISON BETWEEN BIGRAM AND TRIGRAM IN WHOLE ENGLISH DATA SET (3975 WORDS).

	bigram	trigram
correct	3334 (84%)	2900 (73%)
wrong	641 (16%)	1075 (27%)

TABLE III
COMPARISON OF MULTISPELL, ASPELL, MICROSOFT WORD AND GOOGLE FOR ENGLISH.

	MultiSpell	Aspell	Microsoft Word	Google
correct	109 (90%)	105 (87.5%)	105 (87.5%)	106 (88%)
wrong	11 (10%)	15 (12.5%)	15 (12.5%)	14 (12%)

TABLE IV
COMPARISON OF MULTISPELL, ASPELL AND TST FOR THE PORTUGUESE LANGUAGE.

	MultiSpell	TST	Aspell
correct	97 (80%)	78 (65%)	65 (54%)
wrong	23 (20%)	42 (35%)	55 (46%)

B. Evaluation of Portuguese Spelling Correction

The last evaluation was done for the Portuguese language. Bruno and Mário [13] implemented an algorithm using Ternary Search Trees (TST). The authors show experiments in correcting a list of some Portuguese words and comparing their results with Aspell. Here we compared MultiSpell on the whole list (120 Portuguese words) available from their experiments explained in [13], applying our algorithm and comparing it with the Aspell and TST algorithm. Given that MultiWordNet does not provide any Portuguese word senses, we used the dictionary made available from [13] comparing the approaches. Our algorithm succeeded to correct 97 misspelled words (80%), TST succeeded to correct 78 misspelled words (65%) and Aspell succeeded to correct 65 misspelled words (54%) as shown in Table IV and Table VI (in the Appendix).

IV. CONCLUSIONS

In this paper we proposed a language-independent spell-checker that is based on an enhancement of a pure n-gram based model. Furthermore, we presented evaluations on English and Portuguese benchmark data sets of misspelled words. The obtained results outperformed other state-of-the-art methods. In future work, we plan to further optimize the algorithm and data structure used to compute the similarity scores. Furthermore, the algorithm should be tested on data sets for other languages.

Misspelling	Correct Spelling	Aspell	Microsoft word	Google	MultiSpell
gouvener	governor	governor	<u>souvenir</u>	<u>gouverneur</u>	<u>convener</u>
gurantees	guarantee	guarantee	guarantee	guarantee	guarantee
guerrila	(guerilla, guerrilla)	guerrilla	guerrilla	guerrilla	(guerilla, guerrilla)
guerrillas	(guerillas, guerrillas)	guerrillas	guerrillas	guerrillas	(guerillas, guerrillas)
Giuseppe	Giuseppe	Giuseppe	Giuseppe	Giuseppe	Giuseppe
habaeus	(habeas, sabaeus)	habeas	<u>habitués</u>	habeas	<u>sabaeus</u>
hierarcical	hierarchical	hierarchical	hierarchical	hierarchical	hierarchical
heros	heroes	heroes	heroes	heroes	<u>herbs</u>
hypocracy	hypocrisy	hypocrisy	hypocrisy	hypocrisy	hypocrisy
independance	Independence	Independence	-	Independence	Independence
intergration	integration	integration	integration	integration	integration
intrest	interest	interest	interest	interest	interest
Johanine	Johannine	Johannes	Johannes	Johannes	Johannine
judisuary	judiciary	judiciary	judiciary	-	judiciary
kindergarden	kindergarten	kindergarten	kindergarten	kindergarten	kindergarten
knowlegeable	knowledgeable	knowledgeable	knowledgeable	knowledgeable	knowledgeable
labatory	(lavatory, laboratory)	(lavatory, laboratory)	(lavatory, laboratory)	laboratory	(lavatory, laboratory)
lonelyness	loneliness	loneliness	loneliness	loneliness	loneliness
legitamate	legitimate	legitimate	legitimate	legitimate	legitimate
libguistics	linguistics	linguistics	linguistics	linguistics	linguistics
lisence	(license, licence)	licence	<u>silence</u>	licence	licence
mathmatician	mathematician	mathematician	mathematician	mathematician	mathematician
ministry	ministry	ministry	ministry	ministry	ministry
mysogynist	misogynist	misogynist	misogynist	misogynist	misogynist
naturaly	naturally	naturally	naturally	naturally	naturally
ocuntries	countries	countries	countries	countries	countries
paraphernalia	paraphernalia	paraphernalia	paraphernalia	paraphernalia	paraphernalia
Palistian	Palestinian	<u>Alsatain</u>	<u>politian</u>	Palestinian	Palestinian
pamflet	pamphlet	pamphlet	pamphlet	pamphlet	pamphlet
psychic	psychic	psychic	psychic	psychic	psychic
Peloponnes	Peloponnesus	Peloponnes	Peloponnes	Peloponnes	Peloponnesus
personell	personnel	personnel	personnel	personnel	personnel
posseses	possesses	possesses	possesses	possesses	possess
prairy	prairie	<u>priory</u>	prairie	prairie	<u>airy</u>
qutie	(quite, quiet)	quite	quite	<u>cutie</u>	<u>queue</u>
radify	(ratify, ramify)	ratify	ratify	ratify	ramify
recommended	recommended	recommended	recommended	recommended	recommended
reciever	receiver	receiver	receiver	receiver	<u>reliever</u>
reconaissance	reconnaissance	reconnaissance	reconnaissance	reconnaissance	reconnaissance
restauration	restoration	restoration	restoration	restoration	<u>instauration</u>
rigeur	(rigueur, rigour, rigor)	<u>rigger</u>	rigueur	-	(rigueur, rigour)
Saterdag	Saturday	Saturday	Saturday	Saturday	Saturday
scandinavia	Scandinavia	Scandinavia	Scandinavia	Scandinavia	Scandinavia
scaleable	scalable	scalable	-	scalable	scalable
secceeded	(seceded, succeeded)	succeeded	succeeded	seceded	succeeded
sepulchure	(sepulchre, sepulcher)	sepulcher	<u>sepulchered</u>	sepulcher	sepulchre
themselves	themselves	themselves	themselves	themselves	themselves
throught	(thought, through, throughout)	(thought, through)	(thought ,through)	<u>throat</u>	(thought ,through, throughout)
troups	(troupes, troops)	(troupes, troops)	troupes	troops	troops
simultaneous	smultaneous	simultaneous	simultaneous	simultaneous	simultaneous
sincerley	sincerely	sincerely	sincerely	sincerely	sincerely
sophicated	sophisticated	<u>suffocated</u>	<u>supplicated</u>	-	sophisticate
surrended	(surrounded, surrendered)	surrounded	surrender	surrender	surrounded
unforetunately	unfortunately	unfortunately	unfortunately	-	unfortunately
unnecesarily	unnecessarily	unnecessarily	unnecessarily	-	unnecessarily
usally	usually	usually	usually	usually	usually
useing	using	using	using	using	<u>seeing</u>
vaccum	vacuum	vacuum	vacuum	vacuum	vacuum

Correct Form	Spelling Error	TST	Aspell	MultiSpell
ípsilon	ipslon	ípsilon	ípsilon	ípsilon
irisar	irizar	irisar	razar	irisar
irupção	irrupção	-	-	irupção
jeropiga	geropiga	jeropiga	Georgia	jeropiga
juiz	juíz	-	juiz	Juiz
lâmpião	lampeão	lâmpião	sarjeta	campeão
lêem	lêm	lês	lema	lêem
linguista	linguísta	-	linguista	linguista
lisonjear	lisongear	lisonjear	lisonjear	lisonjear
logótipo	logotipo	logo tipo	logo tipo	logótipo
maciço	massiço	mássico	mássico	massudo
majestade	magestade	majestade	majestade	majestade
manjerico	mangerico	manjerico	manjerico	manjerico
manjerona	mangerona	tangerina	tangerina	manjerona
meteorologia	metereologia	meteorologia	meteorologia	meteorologia
miscigenação	miscegenação	miscigenação	miscigenação	miscigenação
nonagésimo	nonagessimo	nonagésimo	nonagésimo	nonagésimo
oceânia	oceania	oceânia	Oceania	oceânia
oficina	ofecina	oficina	oficina	oficina
opróbrio	opróbio	aeróbio	próbio	opróbrio
organograma	organigrama	organograma	-	organograma
paralisar	paralizar	paralisar	paralisar	paralisar
perseverança	preseverança	perseverança	perseverança	perseverance
persuasão	persuação	persuasão	persuasão	persuasão
pirinêus	pirenêus	-	pirinêus	pirinêus
pretensioso	pretencioso	pretensioso	pretensioso	pretensioso
privilégio	previlégio	privilégios	privilégios	privilegios
quadricromia	quadricomia	quadricromia	quadriculai	quadricromia
quadruplicado	quadriplicado	quadruplicado	quadruplicado	quadruplicado
quasimodo	quasimodo	-	quisido	quasimodo
quilo	kilo	quilo	Nilo	dilo
quilograma	kilograma	holograma	holograma	holograma
quilómetro	kilómetro	milímetro	milímetro	quilómetro
quis	quiz	quis	qui	juiz
rainha	raínha	rainha	rainha	rainha
raiz	raíz	-	raiz	raiz
raul	raúl	raul	Raul	raul
rectaguarda	retaguarda	rectaguarda	-	rectaguarda
rédea	rédia	rédea	radia	radia
regurgitar	regurjitar	regurgitar	regurgitar	regurgitar
rejeitar	regeitar	rejeitar	regatar	receitar
requero	requero	requere	requero	requer
réstia	rêstea	réstia	resta	réstia
rubrica	rúbrica	rúbreca	rubrica	rubrica
saem	saiem	saíam	saem	caiem
saloíce	saloice	baloice	saloíce	saloíce
sarjeta	sargeta	sarjeta	sarjeta	Sarjeta
semear	semiar	semear	semear	Semear
suiça	suiça	suiça	suiça	Suíça
supor	supôr	-	supor	Supôs
trânsfuga	transfuga	transfira	transfira	trânsfuga
transpôr	transpor	-	-	transportar
urano	úrano	-	-	grano
ventoinha	ventoínha	ventoinha	ventoinha	ventoinha
verosímil	verosímel	-	-	verosímil
vigilante	vegilante	vigilante	vigilante	vigilante
vôo	voo	-	-	ovo
vultuoso	vultoso	vultuoso	-	vultosos
xadrez	xadrês	xadrez	ladres	xadrez
xamã	chamã	chama	chama	chamá
xelindró	xilindró	cilindro	cilindro	xelindró
zângão	zangão	zangai	-	mangão
zepelin	zeppelin	zepelim	zepelim	zepelin
zoo	zoô	zoo	coo	zoo

REFERENCES

- [1] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys*, 24(4), 377-439, 1992.
- [2] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of ACM*, 7(3):171-176.7, 1964.
- [3] W. Peters, "Lexical Resources," NLP group, Dept. of Comp. Sc., Uni. of Sheffield, 2001.
- [4] R. A. Wagner and M. J. Fisher, "The string to string correction problem," *Journal of Assoc. Comp. Mach.*, 21(1):168-173, 1974.
- [5] A. Stanier, "How accurate is Soundex matching?" *Comp. in Genealogy*, vol. 3:7, 1990.
- [6] C. Fellbaum, "WordNet, an electronical lexical database," Cambridge, MIT Press, 1998.

