ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Comparative Study of Spell Checking Algorithms and Tools

Khyati Shah*
Department of Computer Science.
Shrimad Rajchandra Institute of Management and Computer Application. Bardoli. Gujarat, India.
10mca107@srimca.edu.in.

Mayuri Patel
Department of Computer Science.
Shrimad Rajchandra Institute of Management and Computer Application. Bardoli. Gujarat, India.
10mca064@srimca.edu.in.

Jikitsha Sheth
Assistant Professor.
Shrimad Rajchandra Institute of Management and Computer Application. Bardoli. Gujarat, India.
jikitsha.sheth@srimca.edu.in

Dr. kalpesh Lad
Assistant Professor.
Shrimad Rajchandra Institute of Management and Computer Application. Bardoli. Gujarat, India.
kalpesh.lad@srimca.edu.in

Abstract:In this paper we discussed the spellchecking algorithm. First we discussed about basic error types. In which classes of error, its causing factors and some statistics are discussed. Then the algorithms to correct these errors are discussed.At the end of the section we examine some tools and their environment. And compare the functionality provided by them.

Keywords:spellchecker;algorithm; error; phonetic;spellchecking tools.

I. INTRODUCTION

A. *What is Spell Checker?:*

A spell checker is an application program that focuses the word that may be spelled wrongly. This application may be a standalone tool or can be a part of any large applications. A spell checker is normally made up of two parts, one is the routines for scanning inputted text and extracting words from it and the second one is the algorithm for comparing the words against a known list of correctly spelled words that are extracted from the dictionary.

There are two most common error types, non-word errors and real-word error[1]s. Detection of real word error needs some extra efforts and knowledge compare to detection of non- word errors. Even non-word errors are critical to correct. Number of approaches based on minimum edit distance, n-grams, dice coefficient and similarity key techniques can be used to fulfill this task.

The first part of the spell checker, that is scanning routine normally includes the algorithms to handle the morphology. The generalized work of this routine is, it find out the words having similar pronunciation. This routine can use any of the phonetic algorithms like soundex, metaphone, double metaphone. And the second routine can use the similarity measure algorithms to measure the similarity between the wrongly spelled word and the words which are extracted by the first routine. It can use one of the similarity measure algorithms like edit distance, n-gram, dice coefficient etc.

Spellcheckers are arising to attract the attention of many language and speech applications as they grab towards using online sources for textual input. Spellcheckers can also be used in areas like word processor, email editor, blogs, chat records, electronic dictionary, search engine, data mining. The spellchecker is used as a part of text processing activities in these applications and they are usually known as text cleaning or text normalization.

The quality of text from many sources specifically blogs, emails and chat records may be filled with spelling errors. With the increased use of online resources, spell checking is very important.A Spellchecking activity prevents wasted computational processing also prevents the wastage of user time and efforts and make any system more robust as spelling and typing errors prevent any system form giving the required information.

II. BASIC ERROR PATTERNS

According to Damerau, approximately 80% of all misspelled words contained a single instance of one of these errors: insertion, deletion, substitution, transposition. They are called single error misspellings. A spelling contain more than one such error are called multi-error misspellings.

A. *Word Length Effects:*

From the Zipfs Law[2] it can be conclude that, short words occur more frequently than the long words. Pollock and Zamora studied 50,000 non-word errors, it concludes that errors in short words are difficult for spelling correction, though their frequency of occurrence may be low[3]. They conclude that – although 3-4 character misspellings constitute only 9.2% of total misspellings, they generate 42% of miss corrections. Kukich examinedmore than 2000 error types and conclude that – over 63% of error occurred in words of length 2, 3, 4 characters [4].

B. *First Position Errors:*

It generally supposesthat few errors occur in the first letter of a word. Pollock and Zamora studied 50,000 misspellings and found that 3.3% involved first letter error[3].

© 2010, IJARCS All Rights Reserved

Yannakoudakis and Fawthrop analyzed 1.4% error rate out of 568 typing errors[5]. Mitton found 7% error rate out of 40,000 misspellings that involved first position errors[6]. Kukich also observed 15% first-position error rate[4].

C. Keyboard Effects:

Gruddin analysis for the typing errors, by examine the errors made by 6 expert and 8 beginner typists while copying out magazine articles. Which are totaling about 60000 characters of text[7]. He observed a large gap in typing speed of both and also in type of errors made. According to one observation form that, error rate ranged from 0.4% to 0.9% for experts and 3.2% for beginners approximately. Expert's errors were type of insertions while the majority of beginner's errors were substitutions.

D. *Phonetic Errors:*

Van Berkel and DeSmedt [1988] studied 10 Dutch subjects copya tape recording of 123 Dutch surnames randomly chosen from a telephone directory. They found that 38% of the spellings were incorrect in spite ofbeing phonetically acceptable. Mitton's study 44970 of all the spelling errors in 925 student essays involved homophones.

E. Word Boundary Errors:

There are basically two types of word boundary error: incorrect splits (e.g. *together* ! *together*) and run-ons (e.g. *a lot* ! *alot*). Kukich found that 15% of all errors were word boundary errors. Mitton analyzed that 13% of his 4218 errors were word boundary errors[6].

The following table represents the study relevant to the above discussion.These observations can be explained by the characteristics of the studies: the size of the quantity from which the errors have been taken, the text type, the error types taken into account and the number of errors considered. A question mark indicates that the information is not available for that particular study.

Table: 1 Error Statistics

Study	*Corus size*	*Text type*	*#of Errors*	*Error types*
Wing and Baddeley (1980)	80,000 words	Handwritten essays ofCambridge college applicants	1,185 (1.5%)	No typing errors
Pollock and Zamora (1983 and 1984)	25,000,000 words	Scientific and scholarly text	50,000 (0.2%)	No real-word errors
Yannakoudakis and Fawthrop (1983)	1,014,312 + 60,000 words	Brown corpus and texts written by high-educated bad spellers	1,377 (0.1%)	All sorts of errors
Mitton (1987)	170,016 words	Handwritten student Compositions	4218 (2.5%)	No typing errors
Kukich (1990)	?	Written conversations between deaf people	2,000	All sorts of errors

III. ALGORITHMS

A. Levenshtein Edit Distance:

A levenshtein edit distance is a similarity measure algoritm for two strings. The distance is in terms of the number of deletions, insertions, or substitutions required to transform source string into target string [8]. The Levenshtein distance between the words "kitten" and "sitting" is 3, which can measures as the edits are, **k**itten → **s**itten (substitution of 's' for 'k'), sitt**e**n → sitt**i**n (substitution of 'i' for 'e'), sittin → sittin**g** (insertion of 'g' at the end).The greater the Levenshtein distance indicates the strings are more different.

$f(0,0) = 0$ (1)

$f(i\ ,\ j) = \min[\ (f(\ i-1,j)+1),\ (f(i,j-1)+1),\ (f(i-1.j-1) + d(q_i,l_j))]$ (2)

Where $d(q_i,l_j) = 0$ if $q_i = l_j$, else $d(q_i,l_j) = 1$, [9]

For all strings, a function f(0,0) is set to 0 and then a function f(i,j) is calculated for all query letters, iteratively counting the string difference between the query q1q2 . . . qi and l1l2 . . . lj. Each insertion, deletion, or substitution is awarded a score of 1. Edit distance is O(m,n) for retrieval as it performs instinctiveforce comparison with every character (all m characters) of every word (all n words) in the dictionary. Because of this it can be slow when using large dictionaries.

Levenshtein distance can be calculated by reserve a matrix to hold the Levenshtein distance between all prefixes of the first string and the second string. Matrix isbe flood filled, and thus finds the distance between two strings. The last value computed, at the right most bottom corner shows the distance between two strings. The objective is to find matches for short strings. It can be used where a small number of differences are expected. The cost of finding the difference is roughly the product of length of two strings.Variations of the Levenshtein distance can be acquire by changing the set of operations that can be applied on the strings [edit operations].The damerau – Levenshtein distance supports insertion, deletion, substitution and transposition of two adjacent characters [10], whereas hamming distance only allows substitution and because of it can be applied to the strings of the same length.

This algorithm has a wide range of applications, such as spellcheckers, correction system for OCR [Optical Character Recognition].

An algorithm can be adjust to use less space, o(m) instead of o(m,n). Since it only requires the values of the previous row and current row and current row to be stored at one time. We can also store number of insertion, substitution separately or at the position at they occur. This algorithm parallelizes poorly due to large number of data dependencies. However all cost can be computed parallel and the algorithm can be adapt

© 2010, IJARCS All Rights Reserved

to perform the minimum function to eliminate the dependencies.

B. N-gram:

An n-gram is a substring of length n characters that is derived from a word of length greater than or equal to n. Two different forms of n-gram were normally used, digrams with a length of 2,with the set of two adjacent alphabets, and trigrams with a length of 3. Extra space is padded at the start and end of the word before the generation of bigrams, and two before the generation of trigrams. Thus, the word SUBSTRING will look like,

{ *S, SU, UB, BS, ST, TR, RI, IN, NG, G* }

And the following trigrams [11],

{ **S, *SU, SUB, UBS, BST, STR, TRI, RIN, ING, NG*, G** }

Where "*" indicates the blank space. There are n+1 bigrams and n+2 trigrams for string of length n [12].

This method works based on the number of similar bigrams or trigrams. The number of common bigrams is more if the two words are similar.

Dice's coefficient, is used to measure the similarity over the sets. It is calculates the similarity based on the number of common bigrams. It uses the following formula,

$$s = \frac{2|X \cap Y|}{|X| + |Y|}$$

It takes the common bigrams from both the stings and thendivides it by the summation of length of two strings. For example to calculate the similarity between night and nacht, We would find the set of bigrams in each word are as, {ni,ig,gh,ht} and {na,ac,ch,ht}. Each set has four elements, and the intersection of these two sets has only one element "ht". By applying the formula, we get,

$$s = (2 \cdot 1) / (4 + 4) = 0.25.$$

One possible point of improvement is, this algorithm get confused when there is repeated pairs of bigrams and may not provide appropriate result. Hamming distance is variation of n-gram.

C. Similarity Key Techniques:

The similarity key technique is to find out similarly spelled string by assigning them a key such that similarly spelled words have similar keys. Thus, when a key is computed for erroneous word, it will compared and provide suggestions to similar words in the lexicon. This technique is important because it is time consuming and costly process to compare that erroneous word with every word in the lexicon. Three most popular phonetic algorithms, soundex, metaphone, double metaphone.

a. Soundex:

It is used in phonetic spelling correction applications. It maps the key for misspelling consisting of its first letter followed by a sequence of digits. The following are rules of soundex algorithm to calculate key[13].

a. Keep the first letter (in upper case).
b. Replace these letters with hyphens: *a, e, i, o, u, y, h, w.*
c. Replace the other letters by numbers as follows:

   b,f,p,v : 1
   c,g,j,k,q,s,x,z : 2
   d,t : 3
   l : 4
   m,n : 5
   r : 6

b) Delete adjacent repeats of a number.
c) Delete the hyphens.
d) Keep the first three numbers or pad out with zeros.

For example, Lorry -> L-66- -> L600. It is advisable to apply soundex on the dictionary to select the words that can becompared further. The code is first computed for wrongly spelled word and compared with words in the lexicon. Soundex do not provides facility rank the matched outcomes, instead it simply display all the words to the user.

In soundex words that sound similar mat not always have the same soundex key. For example, Huff (H100) and Hough (H200) are pronounced identically, but have different soundex codes. because the different consonant combinations in English may produce the same sound, the soundex algorithm does not see the names as pronounced the same [14]. Words may sound alike but as they start with a different initial, but have a different soundex code. For example, the names Carriage (C620) and Marriage (M620) have different soundex codes even though they sound alike. Since soundex is based on English pronunciation, while working with other languages like some European names may not sound similar as English. An example is the French name Roux - where the x is silent. While Rue (R000) is pronounced identically to Roux (R200), they will have different soundex keys. Sometimes names that doesn't sound alike may have the same soundex code and this will give false positive results in a soundex search.

b. Metaphone:

Metaphone is also a phonetic algorithm.It is variation of the soundex algorithm. It improves Soundex algorithm by using information about variations and inconsistencies in English spelling and pronunciation to yield a more accurate encoding, which does a better job of matching words and names which sound alike. The principle concept is same as soundex; similar sounding words should share the same keys.Metaphone codes use the 16 consonant symbols 0BFHJKLMNPRSTWXY. The '0' represents "th", 'X' represents "sh" or "ch", and the others represent their Standard English pronunciations. The vowels AEIOU are also used, but only at the beginning of the code [14].

c. Double Metaphone:

The original author later produced a new version of the algorithm, which he named Double Metaphone. It is also phonetic encoding algorithm is the second generation of this algorithm. It is called "Double" because it can return both a primary and alternative code for a string; this is explanations for some ambiguous cases as discussed above as well as for multiple variations of surnames with common heritage. For example, encoding the name "Smith" yields a primary code of *SM0* and an alternative code of *XMT*, while the name "Schmidt" yields a primary code of *XMT* and an alternative

© 2010, IJARCS All Rights Reserved

code of *SMT*--both have *XMT* in common. So it handles conflicting cases efficiently [14].

IV. TOOLS

The following are some web based tools. Their working style and environment is discussed below. Some comparison is also done based on the suggestion given by these tools and also their capability to find out various types of errors.

A. Spellcheckplus :

The working style of the spellcheckplus is shown in the following diagram. It will give you the yellow bordered box at the wrongly spelled spelling after pressing the check text button. At the mouse hover, it will show you the suggestion of right word and also give one example related to that word. After pressing modify button, it will replace that wrongly spelled word with the suggested word. For suggestion it usespop-ups.

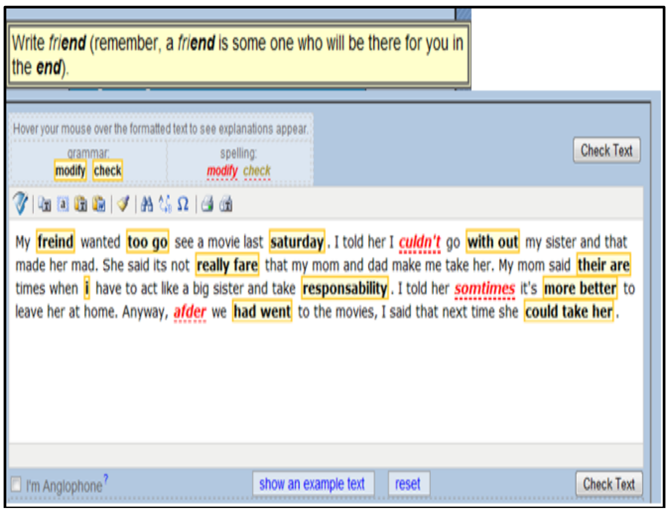

Figure: 1 spellcheckplusenvironment– www.spellcheckplus.com

B. Jspell:

In jspell, it shows error in the text when "SpellCheck" button is pressed. It shows dialog box containing all the possible correct spellings with the options: "Replace, Replace All, Ignore, Ignore All, Learn, and Finish". It shows all the spelling mistakes one by one by selecting the incorrect word.

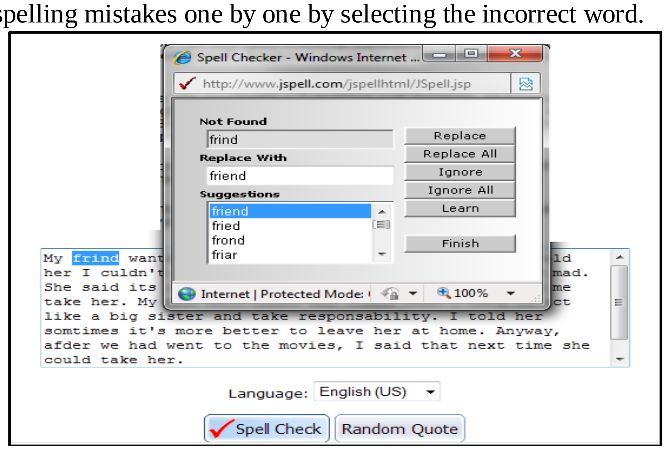

Figure: 2 jspell environment- form www.jspell.com

C. Spelljax:

The working of spelljax is shown below; It indicates the error in the word by red font and underlined that word. After writing the whole text, when we press "Check Spelling" link at the top of the box. It will start editing the whole text. When we press resume editing, it will stop editing the text further. For the suggestion for wrongly spelled word, right click on that word and select appropriate one and the color of the rightly replaced word will be green.

Figure: 3 spelljex environment - from www.spelljex.com

D. ORANGOO:

Figure: 4 orange environment– from www.orangoo.com

Orangoo is another web based spell checker. When you write the text in the given window and press the spell check button given at the bottom of the window, it will open the new window and display entered text with the error highlight. And also provide suggestion for wrongly spelled word and display all suggestion in the given suggestion box. It also provides the facility like ignore, replace, add, and change all and many more.

© 2010, IJARCS All Rights Reserved

E. *CheckDog:*

Figure: 5 Checkdog environment– from www.CheckDog.com

This tool works slight differently compare to other tools. It is in waiting for your text, at the moment you entered the text, it display all the words having the error. When you mouse hover to the suggestion given in the right panel, it will highlight the related word with the red color in the text window. If you select the suggested word,it will replace it.

F. *Respelt:*

As given in the below figure, the respelt provides basically two functionalities. We can provide the url of our website and check the content of our website. Or we can enter the text and check spellings. When we type the text and press check spelling button, it will open the new window with the text entered, and highlight the wrong words with the red color. If you over the mouse on those spellings, it will give you the suggestion for it, from which you can select the appropriate one.

Figure: 6 Respelt environment – from www.respelt.com

Table: 2 Tools comparison table

Description		***Spell Check Plus***	***Spelljax***	***JSpell***	***Orangoo***	***Respelt***	***CheckDog***
Additional explanation		Yes	No	No	No	No	Yes
Capitalization	Capitalization of “I”	Yes	No	No	No	No	Yes
	Other	Yes	Yes	Yes	Yes	Yes	Yes
Grammar		Yes	Yes	Yes	Yes	No	Yes
Punctuation		Yes	No	No	Yes	No	Yes
Grammar Comparison Property [comparative, superlative]		Yes	No	No	Yes	No	Yes
Sentence construction error with verb, noun and pronoun		Yes	No	No	Yes	No	Yes
Sentence formation error according to meaning		Yes	No	No	Yes	No	Yes
Writing past perfect tense instead of present perfect tense		Yes	No	No	No	No	No
Thesaurus Checking		No	No	No	Yes	No	No
Conflict in use of subject and verb		No	No	No	Yes	No	No
Spell Checking of Website		No	No	No	Yes	Yes	Yes

V. CONCLUSION AND FUTURE WORK

In present paper we discussed about all the existing algorithms and tools. And also identify their strength and weakness.We also compare all the tools from different aspects. And conclude that which algorithm gives better result in which criteria. And which additional functionality should be added in existing tool.

In future we can use these facts and will prepare such an algorithm that overcome weakness of the existing algorithms. And we can also develop a tool based on this algorithm.

VI. REFERENCE

[1]. Dr.Hussains,”spell checking”,www.crulp.org.

[2]. Zipf, G. K.,"The Psychobiology of Language". Houghton Mifflin, Boston 1935.

[3]. Pollock, J. J., and Zamora, A., "Collection and characterization of Spelling errors in scientific and scholarly text", J. Amer. Soc. Inf. Sci., pp.51-58,1983.

[4]. Kukich K., "Techniques for automatically correcting words in text", ACM Computing Surveys, pp. 377-439, 1992.

[5]. Yannakoudakis, E. J., and Fawthrop, D.,"An intelligent spelling corrector", Inf. Process. Manage., pp. 101 108, 1983.

[6]. Roger Mitton, "Spelling checkers, spelling correctors and the misspellings ofpoor spellers", Information Processing and Management: an International Journal, pp. 495-505 1987.

[7]. Grudin, J., "Error patterns in skilled and novice transcription typing", In Cognitive Aspects ofSkilled Typewriting, pp. 121-143,1983.

[8]. V.I. Levenshtein,"Binary codes capable of correcting deletions, insertions and reversals". Sov. Phys. Dokl. , pp. 707-710, 1966.

[9]. Hodge v., Austin j., “A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach”, ieee transactions on knowledge and data engineering, vol. 15, no. 5, september/october 2003.

[10]. Damerau, F., ‘A technique for computer detection and correction of spelling errors’, Communications of the ACM 7(3), 171–176,1964.

[11]. Verberne S., “Context-sensitive spell checking based on word trigram probabilities”, Master thesis Taal, February. August 2002.

[12]. Robertson M., Willett p. “Searching For Historical Word-Forms In A Database Of 17th-CenturyEnglish Text Using Spelling-Correction Methods”, Department of Information Studies, University of Sheffield, S1O 2TN.

[13]. Odell, M. & Russell, R., U.s. patent numbers 1,261,167. U.S. Patent Office, Washington, D.C,1918.

[14]. Website:www.wikipedia.org/

© 2010, IJARCS All Rights Reserved