A Novel Approach for Phonetic Based Stem Generation Using Edit Distance and Longest Common Subsequences

Mrs.M.Kasthuri,
Asst.Professor, Dept. of Com.Science,
Bishop Heber College (Autonomous),
Tiruchirappalli, Tamil Nadu, India.
kasthuri_saro@yahoo.com

Dr.S.Britto Ramesh Kumar,
Asst.Professor, Dept. of Com.Science,
St.Joseph's College (Autonomous),
Tiruchirappalli, Tamil Nadu, India.
brittork@gmail.com

***ABSTRACT-* In the present work an innovative attempt is being made to develop a novel conflation method that exploits the phonetic quality of words and uses some standard NLP tools like LD (Levenshtein Distance) and LCS (Longest Common Subsequence) for Stemming process.**

Keywords:
Phonetic based stem generation system.

I. INTRODUCTION

In linguistic morphology, stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form– generally a written word form. One technique for improving IR performance is t o provide searchers with ways of finding morphological variants of search terms.

II. RELATED WORKS:

Several stemming algorithms have been suggested in the literature, which can be classified as affix removal methods, statistical methods and lexicon based (or mixed) methods. Lovin's al gor it hm [8], Port er's al gorit hm [14], Paise/Husk st emmer [13] and Harman's 'S' st emmer [6] ar e some of t he affix removal techniques. These algorithms apply a set of rules, typically known as transformation rules applied to each word. These algorithms are language specific and depend on priory knowledge of language morphology. Statistical algorithms try to cope with this problem by finding distributions of root elements in a corpus but they require rich computing resources as heavy computations are necessary for such approaches. Some highlighted statistical stemming algorithms available in the literature are: Successor Variety [5], Corpus based Stemming [19], N-gram Stemming [9], HMM based stemmer [11], YASS [10]. The third type of stemming utilizes the inclusion of dictionary lookups. The strength of such approaches is in their ability to produce morphologically correct stems but the major and obvious flaw in dictionary-based algorithms is their inability to cope with words, which are not in the lexicon. It is also true that a lexicon must be manually created in advance, which requires si gnif icant effort s. Robert Krovet z's st emmer [7] is one the example of such approach. Some interesting variations to stemming can be seen in recent years. These are Joshua S. English [4], Eiman Tamah, Al- Shammari [17], J. Šn ajder et al. [15], Lourdes Araujo et al. [1]. The current method takes into account how the words are being pronounced, thereby utilizing the phonetic quality of words and apply a simple set of rules to achieve the desired stem.

III. PROPOSED METHOD

For a given word, our idea is to collect words that share same phonetic code first, and then apply some sort of filtering to acquire the most appropriate stem among them. We hypothesize a phonetic based stem generation system.

A. Phonetic Algorithm

Phonetic algorithms encode words based on their pronunciation. In his article, Brijesh Shankar Singh [16] has advised that Soundex algorithm [12] (a name matching algorithm based on 6 phonetic sound classifications) could be used for removing affixes. This comment has inspired us to use Metaphone algorithm [2], the advancement of Soundex as a tool to develop a stemming system. Naushad UzZaman and Mumit Khan [18] have presented an extension of T9 system (Text on 9 keys, a predictive text technology used in cell phones) called T12. In this text input system, the author used the phonetic encoding; the metaphone code to gather words corresponding to a particular sequence of key hits. The initial tasks of our system is to generate the metaphone code or key and collect words that has same metphone code up to 4 characters, as the first four letters of the phonetic spelling (if there are that many) are used for comparisons in standard traditional Metaphone algorithm.

B. Similarity Measures

For filtration, the input word needs to be matched against all words in hand. We shall use two NLP tools for this purpose. Levenshtein distance (LD) or Edit distance (ED), available online at http://www.merriampark.com/ld.htm is a measure of the similarity between two strings. The distance is the number of deletions, insertions, or substitutions required to transform the source string (S) into the target string (T). The Levenshtein

ISSN: 2278-0181

distance algorithm has been used in Spell checking, Speech recognition, DNA analysis, and Plagiarism detection. The longest common subsequence (LCS) [3] is the longest subsequence common to all sequences in two strings. It is a classic computer science problem, the basis of diff (a file comparison program that outputs the differences between two files), and has applications in bioinformatics.

C. *Rule Generation*

Having two metrics in hand, we have devised two simple rules to filter out the word list. Suggestion of both the rules has an empirical base. Considering the following words with their EDs and LCSs with the word Superconductivity, following table can be generated:

Table 1. EDs and LCSs Values with the word superconductivity

Word	Ed	LCS
Supercargo	11	7
Supercharge	11	6
Supercoil	9	8
Superconduct	5	12
Superconduction	4	13
Superconductive	3	14
Superconductivit y	0	17
Superconductor	5	12
Supercontinent	7	11
Supercool	10	7

If we draw these points in XY plane, the graphical representation would be:

The equation of the line thus plotted is

$$(y-y1) = \frac{(y2-y1)}{(x2-x1)} (x-x1)$$

$$(y-12) = \frac{(13-12)}{(4-5)} (x-5)$$

$$(y-12) = (-1)(x-5)$$

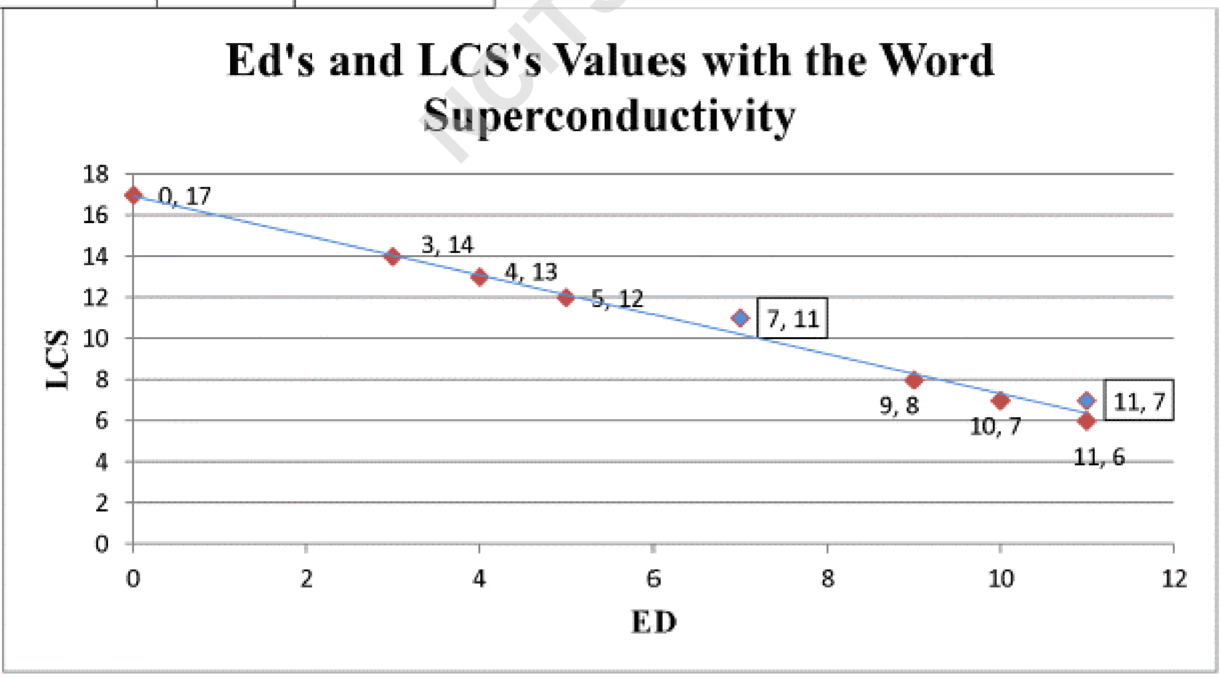
$$x+y = 17 \quad \ldots. (3.3.1)$$

Also, the length or string size of the input word Superconductivity,

[SUPERCONDUCTIVITY]=17

This makes us to infer that,

ED + LCS = Input word length …. (3.3.2)



It is obvious that the points that do not satisfy the equation (3.3.1) shall fall outside the line, so we shall rule out the words for which (3.3.2) does not hold. Again, we can see that still there are some words (supercharge, supercoil and supercool) that need to be excluded. For this, we reason as following:

The pair (ED, LCS) can satisfy the rule (3.3.2) with following three possibilities:

ISSN: 2278-0181

i. ED>LCS

ii. ED=LCS

iii. ED<LCS

The condition (i) implies that the number of operations required to convert string A to string B is greater than what is common between them. Consequently there is very low probability that A is the stem of B. Condition (ii) shows that there are as many number of operations to convert string A to string B as what is common between them. As such a relative higher value of LCS has the same high ED value associated with it. So, we may think of a situation where we have to cover some larger distance from the end point of common sub string to the end of target string to match string A to string B, no matter how large sub string they have in common. Condition (iii) implies that the numbers of operations to convert string A to string B are less then what they have common. This shows an ideal situation where two strings have relatively higher common sub sequence and there are few operations needed to convert string A to string B. So, there is high probability that string A is the stem of B. On the basis of above justification, the other rule for filtration can be accepted as:

ED < LCS .... (3.3.3)

D. Selection of stem

Stem is the valid word which is of shortest length among all of its morphological variants. Thus, we select the word for which ED is the maximum and corresponding LCS is the minimum from the filtered list of words. If there is more than one word, then we choose the word as stem whose length is the minimum.

IV. THE ALGORITHM

The process shall be performed in following steps:

A. Accept the input word.

B. Generate its Metaphone Code.

C. Find the list of words that have the same Metaphone code as the input word up to 4 characters.

D. Calculate the Edit Distance (ED) and Longest Common Subsequence (LCS) for each word with respect to the input word.

E. Rule out the words for which following does not hold:
   a. ED + LCS = Input word length
   b. ED < LCS

F. Select the word(s) for which ED is maximum and corresponding LCS is the minimum.

G. If there is more than one word, then choose the word as stem whose length is the minimum.

V. ILLSTRATIVE EXAMPLES

Suppose we have to find the stem of the word

Superconductivity.

1. Input word= Superconductivity

2. Metaphone Code= SPRKNTKFT

3. Words with the code SPRK:

   i. Supercargo (SPRKRK)
   ii. Supercharge (SPRKRJ)
   iii. Supercoil (SPRKL)
   iv. Superconduct (SPRKNTKT)
   v. Superconduction (SPRKNTKXN)
   vi. Superconductive (SPRKNTKTF)
   vii. Superconductivity (SPRKNTKFT)
   viii. Superconductor (SPRKNTKTR)
   ix. Supercontinent (SPRKNTNNT)
   x. Supercool (SPRKL)

4. EDs and LCSs for these words are shown in following

Table 2.

5. Checking of rule is shown in following Table 2.

Table 2. Eds and LCSs values and Rule Follows with Superconductivity

Word	Ed with word superconductivity	LCS with word superconductivity	Rule Follow?	
			I	II
Supercargo	11	7	N	N
Supercharge	11	6	Y	N
Supercoil	9	8	Y	N
Superconduct	5	12	Y	Y
Superconduction	4	13	Y	Y
Superconductive	3	14	Y	Y

ISSN: 2278-0181

Superconductivity	0	17	Y	Y
Superconductor	5	12	Y	Y
Supercontinent	7	11	N	Y
Supercool	10	7	Y	N

Since for the words Supercargo and Supercontinent Rule (I) is violated, so we shall drop these words. Similarly, for the words Supercharge, Supercoil and Supercool Rule (II) has violated, so we shall drop these words too.

6. Out of the remaining words, word(s) for which ED is maximum and LCS is minimum:

Superconduct (5, 12)

Superconductor (5, 12)

7. Since there are more than one word, and length of Superconduct is less than Superconductor, i.e.

[SUPERCONDUCT] < [SUPERCONDUCTOR]

So we select Superconduct as the stem of the input word Superconductivity.

VI. CONCLUSION

We have proposed a novel phonetic based stem generation system that exploits the phonetic quality of words. The present system, based on phonetic quality can handle misspelled input words and produces morphologically correct word as stem. Actual application of the designed algorithm to the testing data gave nearly 100 percent results in terms of producing correct stems.

VII. FUTURE ENHANCEMENT

The used phonetic algorithm can be extended to Double Metaphone or Metaphone 3. Such type of research work paves the way for using more than 4 characters of phonetic spelling which in turn can generate better results.

VIII. REFERENCES

[1] Araujo Lourdes, Zaragoza Hugo, Pérez-Agüera Jose R., Pérez-Iglesias Joaquín (2010) Structure of morphologically expanded queries: A genetic algorithm approach, Data & Knowledge Engineering, Volume 69, Issue 3, 279-289

[2] Binstock A. and Rex J. (1995) Practical Algorithms for Programmers, Addison-Wesley, Reading, Mass., 158-160

[3] Cormen T. T. , Leiserson C. E. , Rivest R. L. (1990) Introduction to algorithms, MIT Press, Cambridge, MA [4] English Joshua S. (2005) English Stemming Algorithm, Pragmatic Solutions, Inc., 1-3

[4] Hafer M., and S. Weiss (1974) Word Segmentation by Letter Successor Varieties, Information Storage and Retrieval, 10, 371-85.

[5] Harman Donna (1991) How effective is suffixing?, Journal of the American Society for Information Science, 42, 7-15

[6] Krovetz Robert (1993) Viewing morphology as an inference process, Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, 191-202

[7] Lovins, J. B. (1968) Development of a stemming algorithm, Mechanical Translation and Computational Linguistics, 11, 22-31

[8] Mayfield James and McNamee Paul (2003) Single Ngram stemming, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 415-416

[9] Majumder Prasenjit, Mitra Mandar, Parui Swapan K., Kole Gobinda, Mitra Pabitra and Datta Kalyankumar (2007) YASS: Yet another suffix stripper, ACM Transactions on Information Systems. Volume 25, Issue 4, Article No. 18

[10] Melucci Massimo and Orio Nicola (2003) A novel method for stemmer generation based on hidden Markov models, Proceedings of the twelfth international conference on Information and knowledge management, 131-138

[11] Odell K. M. and Russell R. C., Soundex phonetic comparison system [cf.U.S. Patents 1261167 (1918), 1435663 (1922)].

[12] Paice Chris D (1990) Another stemmer, ACM SIGIR Forum, Volume 24, No. 3. 56-61.

[13] Porter M.F (1980) An algorithm for suffix stripping, Program 14, 130-137

[14] Šnajd er J. , Baši c B. Dalbelo, Tadi c M. (2008) Automatic acquisition of inflectional lexica for morphological normalization, Information Processing & Management, Volume 44, Issue 5, 1720-1731

[15] Singh Brijesh Shanker (2003) Search Algorithms, DRTC Workshop on Digital Libraries: Theory and Practice, Paper: E

[16] Tamah Eiman , Shammari-Al. (2008) Towards an error free stemming, IADIS European Conference Data Mining, 160-163

[17] UzZaman Naushad and Khan Mumit, (2005) T12: An Advanced Text Input System with Phonetic Support for Mobile Devices, 2nd International Conference on Mobile Technology, Applications and Systems, 1-7.

[18] Xu Jinxi and Croft Bruce W (1998) Corpus-based stemming using co-occurrence of word variants, ACM Transactions on Information Systems. Volume 16 (1)1, 61-81.