ADVANCEMENT OF CLINICAL STEMMER

Pramod Premdas Sukhadeve[1] and Dr. Sanjay Kumar Dwivedi[2]

[1]Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Luck now, India
Sukhadeve.pramod@gmail.com
[2]Department of Computer Science, Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar Raebareli Road, Lucknow, India
Skd2000@yahoo.com

Abstract:

Word Stemming is common form of language processing in most Information Retrieval (IR) systems. Word stemming is an important feature supported by present day indexing and search systems. Idea is to improve by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching. Stemming is usually done by removing any attached suffixes, and prefixes from index terms before the assignment of the term. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the number of retrieved documents. Texts from the medical domain are an important task for natural language processing. This paper investigates the usefulness of a large medical database for the translation of medical documents using a rule based machine translation system. We are able to show that the extraction of affixes from the words.

Keywords: Stemming, Information Retrieval, Suffix, Prefix, Natural Language Processing.

Introduction:

Stemming is the procedure of finding the root word, by stripping away the affix attached to the word. In many languages words are often obtained by affixing existing words or roots. Stemming is a widespread form of language processing in most information retrieval systems [1]. It is similar to the morphological process used in natural language processing, but has somewhat different aims. In an Information retrieval system, stemming is used to reduce different word forms to common roots, and thereby improve the aptitude of the system to match query and document vocabulary. It also helps in clinical language to knob the clinical terms, names of deceases and symptoms of patient. Although stemming has been studied mainly for English, there is evidence that it is useful for a number of languages. Stemming in English is usually done during document indexing by removing word endings or suffixes using tables of common endings and heuristics about when it is appropriate to remove them. Thus using a stemmer improves the number of documents retrieved in response to translate the clinical data. Also, since many terms are mapped to one, stemming serves to decrease the size of the index files in the information retrieval system. Many stemming algorithms have been proposed, and there have been many experimental evaluations of these. But, very few work on stemming has been reported for clinical language. This paper investigates the usefulness of a large medical database for the translation of documents; we present a stemmer for clinical language. This conflates[1] terms by stripping off word endings from a suffix list maintained in a database.

---

[1]The term conflates is used to denote the act of mapping variants of a word to a single term or 'stem'.

English Stemming Word curriculum

The first step while developing a stemmer is to define the word curriculum and the grammatical information that will be required for words of these word classes natural language processing application for that language. After significant of word classes for English and the grammatical information that is required from the words of these word classes, various paradigms for these word classes were developed. Paradigm for a root word gives information about its achievable word forms in a particular word class, and their relevant grammatical information. All the words of a word class may not follow the same paradigm, like; it is not that all nouns will follow the same inflectional pattern. So, the first assignment was to find out the various paradigms for a word class and then group the words of that word class according to those paradigms. Proceeding this way paradigms were developed for the word classes which show inflection. For developing the paradigms the inflectional patterns of the root words of a word class were studied. And, then on their basis, the root words which inflect in the similar way were grouped. The inflection patterns for those groups constitute the set of paradigms for that word classes. Following is the list of word classes along with their grammatical information that are being used for English.

Noun	Grammatical information required for English is –Number, gender, type, and syntactic features. Nouns have singular and plural forms. Many plural forms have -s or -es endings (dog/dogs, referee/referees), in English, nouns do not have grammatical gender. However, many nouns can refer to masculine or feminine animate objects (mother/father, tiger/tigress, male/female). Nouns have several syntactic features that can aid in their identification. The natural language English has noun which indicates the name of the persons, things, etc.	Nouns (example: common noun "cat") may be modified by adjectives ("the beautiful Angora cat"), preceded by determiners ("the beautiful Angora cat"), or pre-modified by other nouns ("the beautiful Angora cat").
Verb	Verb form the second largest word class after nouns. According to Carter and McCarthy, verbs denote "actions, events, processes, and states." Consequently, "smile," "stab," "climb," "confront," "liquefy," "wake," "reflect" are all verbs. verb is used to describe the action or activity of noun.	Some examples of verb endings, which while not dead giveaways, are often associated, include: "-ate" ("formulate"), "-iate" ("inebriate"), "-ify" ("electrify"), and "-ize" ("sermonize"). There are exceptions, of course: "chocolate" is a noun, "immediate" is an adjective, "prize" can be a noun, and "maize" is a noun. Prefixes can also be used to create new verbs. Examples are: "un-" ("unmask"), "out-" ("outlast"), "over-" ("overtake"), and "under-" ("undervalue"). Just as nouns can be formed from verbs by conversion, the reverse is also possible.
Adjectives	Adjectives describe properties, qualities, and states attributed to a noun or a pronoun. As was the case with nouns and verbs, the class of adjectives cannot be identified by the forms of its constituents. However, adjectives are commonly formed by adding the some suffixes to nouns.	Examples: "-al" ("habitual," "multidimensional," "visceral"), "-ful" ("blissful," "pitiful," "woeful"), "-ic" ("atomic," "gigantic," "pedantic"), "-ish" ("impish," "peckish," "youngish"), "-ous" ("fabulous," "hazardous"). Adjectives can also be formed from other adjectives through the addition of a suffix or more commonly a prefix: weakish, implacable, disloyal, irredeemable, and unforeseen. A number of adjectives are formed by adding "a" as a prefix to a verb: "adrift," "astride," "awry."
Adverb	Adverbs are a class of words "which perform a wide range of functions. Adverbs are especially important for indicating time, manner, place, degree, and frequency of an event, action, or process." They typically modify verbs, adjectives, or other adverbs. Adjectives and adverbs are often derived from the same word. A majority of adverbs are formed by adding to "-ly" ending to their corresponding adjective form. Recall the adjectives, "habitual", "pitiful", "impish".	Some suffixes that are commonly found in adverbs are "-ward(s)" and "-wise": "homeward": "The ploughman homeward plods his weary way." "downward": "In tumbling turning, clustering loops, straight downward falling, ..." "lengthwise": 2 to 3 medium carrots, peeled, halved lengthwise, and cut into 1-inch pieces.

Table 1. Delineate of Grammatical segment

Stemmers are used to convert inflected words into their root or stem. Stem does not necessarily correspond to linguistic root of a word. Stemming improve performance by reducing morphologically variants into same words. There are few rules when using medical roots, "o" always acts as a joint-stem to connect two consonantal roots, e.g. *arthr+o+logy= arthrology.* But generally, the "o" is dropped when connecting to a vowel stem, e.g. *arthr+itis=arthritis, instead of arthr-o-itis.*
The list of some roots, suffixes and prefixes used in medical terminology are shown below in table 1.

Words	Prefix	Suffix	Stem/Root Words
Treatment	------	-ment	Treat
Illness	------	-ness	Ill
Stitching	St	-ing	Itch
Hypogastria	Hypo	-tria	Gas
Abortion	------	-tion	Abort
Abscesses	------	-es	Abscess
Hypertension	Hyper	-sion	Tense

Table 1. Root words of Clinical Terminology.

Rules for Suffix
There are certain rules for suffix of the words ending with 'able','ment','ing',etc… the rules are as follows

1. Rules for suffix 'able' are as follows

a) If in a word before 'able', 'b' comes with vowel 'i' then replace 'able' by 'e'
Example, describable ⟶ descr**ib**+able ⟶ describe
ascribable ⟶ ascr**ib**+ able ⟶ ascribe

b) If in a word before 'able', 'b' comes with any consonant or vowel (expet 'b') then remove 'able'.
Example, absorbable ⟶ abso**rb**+able ⟶ absorb
climbable ⟶ cli**mb**+ able ⟶ climb

c) If in a word before 'able', 'h' comes with any consonant or vowel then remove 'able'
Example, abolishable ⟶ abolis**h**+able ⟶ abolish
accomplishable ⟶ accomplis**h**+able ⟶ accomplish

2. Rules for suffix 'ment' are as follows

a) If in a word suffix'ment' comes then remove 'ment'.
Example, abandonment ⟶ abandon+ment ⟶ abandon
establishment ⟶ establish+ment ⟶ establish

3. Rules for suffix 'ly' are as follows

a) If in a word suffix 'ly' comes then remove 'ly'.
Example, kindly ⟶ kind+ly ⟶ kind
softly ⟶ soft+ly ⟶ soft

4. Rules for suffix 'ness' are as follows

a) If in a word suffix 'ness' comes then remove 'ness'.
Example, cleverness ⟶ clever+ ness ⟶ clever
darkness ⟶ dark+ness ⟶ dark

Rules for Prefix
There are certain prefixes such as dis, im, in, mis, pre, re, un,…etc rules for prefix is shown below

a) If in a word prefix 'dis' comes then remove 'dis' from the word

Examples, disagree ⟶ dis+agree ⟶ agree
disorder ⟶ dis+order ⟶ order

b) If in a word prefix 'im' comes then remove 'im' from the word.
Example, impatient ⟶ im+patient ⟶ patient
Impossible ⟶ im+possible ⟶ possible

Existing work on stemmer

Documents are generally represented in terms of the words they contain, as in the vector-space model [2]. Many of these words are similar to each other in the sense that they denote the same concept(s), i.e., they are semantically similar. Generally, morphologically similar words have similar semantic interpretations, although there are several exceptions to this, and may be considered equivalent. The construction of such equivalence classes is known as stemming. A number of stemming algorithms or stemmers, which attempt to reduce a word to its stem or root form, have been developed. Thus, the document may now be represented by the stems rather than by the original words. As the variants of a term are now conflated to a single representative form, it also reduces the dictionary size, which is the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in savings in storage space and processing time.

Stemming is often used in information retrieval because of the various advantages it provides [3]. The literature is divided on this aspect, with some authors finding stemming helpful for retrieval tasks [3], while others did not find any advantage [4]. However, they are all unanimous regarding the other advantages of stemming. Not only is the storage space for the corpus and retrieval times reduced but recall is also increased without much loss of precision. Moreover, the system has the option for query expansion to help a user refine his/her query.

Different Stemming Algorithms

Various stemmers are available for several languages, including English. The most prominent ones are those introduced by Lovins, Dawson, Porter, Krovetz, Paice/Husk and Xu, and Croft. We now provide a brief description of some of these algorithms.

1. Truncate(n): This is a trivial stemmer that stems any word to the first n letters. It is also referred to as n-gram stemmer [5]. This is a very strong stemmer. However, when n is small, e.g., one or two, the number of overstemming errors is huge. For this reason, it is mainly of academic interest only. In this paper, we have chosen n to be 3, 4, and 5 and refer to them as trunc3, trunc4 and trunc5, respectively.
2. Lovins Stemmer: The Lovins stemmer [6] was developed by Lovins and is a single-pass longest match stemmer. It performs a lookup on a table of 294 endings, which have been arranged on a longest match principle. The Lovins stemmer removes the longest suffix from a word. Once the ending is removed, the word is recoded using a different table that makes various adjustments to convert these stems into valid words. However, it is highly unreliable and frequently fails to form words from the stems or to match the stems of like-meaning words.
3. Dawson Stemmer: The Dawson stemmer [7], which was developed by Dawson, extends the Lovins stemmer. This is also a single-pass longest match algorithm, but it uses a much more comprehensive list of around 1200 suffixes, which were organized as a set of branched character trees for rapid access. In this case, there is no recoding stage, which had been found to be unreliable.
4. Porter Stemmer: Porter proposed the Porter stemmer [8], which is based on the idea that the suffixes in the English language (approximately 1200) are mostly made up of a combination of smaller and simpler suffixes. It has five steps, and within each step, rules are applied until one of them passes the conditions. If a rule is accepted, the suffix is removed accordingly, and the next step is performed. The resultant stem at the end of the fifth step is returned.
5. Paice/Husk Stemmer: The Paice/Husk stemmer [9] is a simple iterative stemmer and uses just one table of rules; each rule may specify either deletion or replacement of an ending. The rules are grouped

into sections that correspond to the final letter of the suffix, making the access to the rule table quicker. Within each section, the order of the rules is significant. Some rules are restricted to words from which no ending has yet been removed. After a rule has been applied, processing may be allowed to continue iteratively or may be terminated.

6. Krovetz Stemmer: The Krovetz stemmer [10] was developed by Krovetz and makes use of inflectional linguistic morphology. It effectively and accurately removes inflectional suffixes in three steps: the conversion of a plural to its singular form, the conversion of past to present tense, and the removal of -ing. The conversion process first removes the suffix and then through the process of checking in a dictionary for any recoding, returns the stem to a word. It is a light stemmer in comparison to the Porter and Paice/Husk stemmers.

7. Co-Occurrence-Based Stemmer by Xu and Croft: Xu and Croft [5] observed that most stemmers perform understemming or overstemming, or even both. Strong stemmers generally perform overstemming only. Xu and Croft came up with an algorithm that would refine the stemming performed by a strong stemmer. To this end, they computed the co-occurrences of pairs of words that belong to the same equivalence class. For each pair, they also computed the expected number of co-occurrences, which would account for words that occur together randomly. Thus, they obtained a measure that is similar to the mutual information measure

8. Dictionary-Based Stemmers: There have also been dictionary-based stemmers [3], [11], [12] that improve on an existing stemmer by employing knowledge obtained from a dictionary. Word co-occurrences in a dictionary are considered to imply the relations between words.

9. Probabilistic Stemmers: Given a word in a corpus, the most likely suffix–prefix pair that constitutes the word is computed [13]. Each word is assumed to be made up of a stem (suffix) and a derivation (prefix), and the joint probability of the (stem, derivation) pair is maximized over all possible pairs constituting the word. The suffix and prefix are chosen to be nonempty substrings of the given word, and it is not clear what should be done in the case when a word should be stemmed to itself.

10. Refinement of an Existing Stemmer: In some cases, errors produced by a stemmer are manually rectified by providing an exception list [10]. The stemmer would first look up the exception list, and if the word is found there, it returns the stem found there. Otherwise, it uses the usual stemmer. The aforementioned co-occurrence-based stemmer is also one such algorithm where the exceptions are obtained automatically.

11. Distributional Clustering as Stemming: Distributional clustering [14], [15]–[16] joins (distributionally) similar words into a group if the words have similar probability distributions among the target features that co-occur with them. In the distributions are estimated by observing the grammatical relationships between words and their contexts, whereas, the distributions are obtained from the frequency of words in each category of the corpus. In their work on document classification, Baker and McCallum had chosen the class labels as the target features. The root forms of the words are not taken into consideration while grouping them. This algorithm described is given as follows. The mutual information of each word in the corpus with the class variable is computed, and the words are sorted in descending order. The number of desired clusters is fixed beforehand, e.g., to M. The first M words are initialized to form M singleton clusters. The two most similar (of the M) clusters are merged. This similarity is measured in terms of the Kullback–Leibler divergence of the distributions of the two clusters. The next word in the sorted list forms a new singleton cluster. Thus, the number of clusters remains M each time. In this paper, we refer to Baker and McCallum's method as baker. In our implementation, we have fixed M to the number of stems obtained by refining the trunc3 stemmer using our model.

Features of Proposed Stemmer

Stemmer for clinical language has windows platform. It has unproblematic to use GUI (Graphical User Interface) for the user to operate and need not to have much knowledge about computers, platforms and any programming language. Users just need some essential computer operation knowledge for software installation and manoeuvre. If we confer from the technical point of view, it has been developed using Visual Basic as Front End and Oracle10g as Back End. It is easy to use and give accurate root words. The

proposed stemmer may be useful in medical field which is usually done during document indexing by removing word endings or suffixes using tables of common endings and heuristics about when it is appropriate to eliminate them. Following stature shows the stream of words in database.

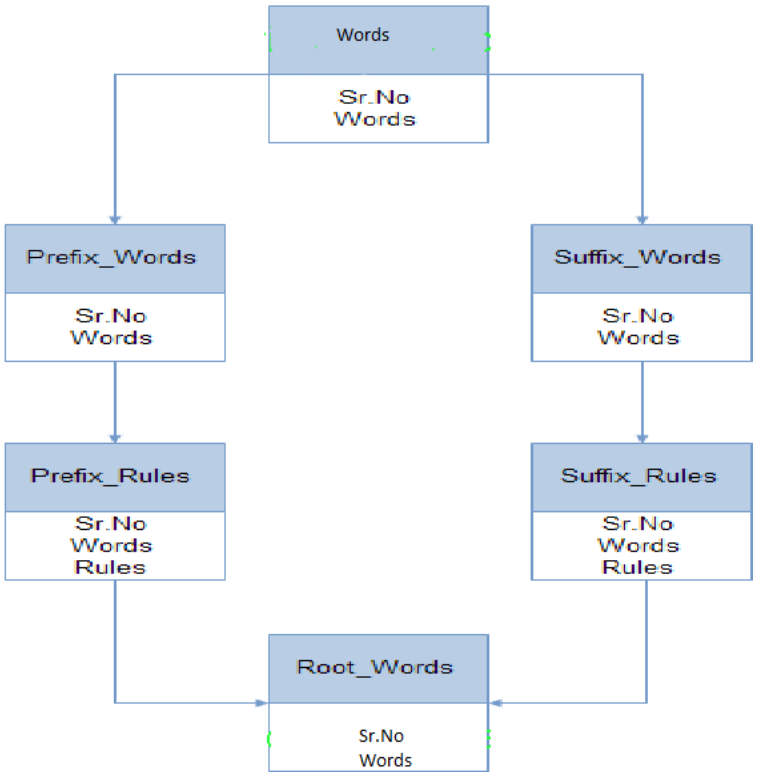

Figure 2. UML diagram of database

We need to develop a new stemmer because the active stemmers which are Algorithm based are not able to give correct root words in some of the words The stemmers are completely based on general languages (regional, communicative), but the clinical terminology is someway diverse from the wide-ranging languages, and the stemmers which are database based are not equipped to give the proper root words of clinical terminology. Consequently we have urbanized the new stemmer based on database which will bestow the appropriate output.

Conclusion

The English clinical stemmer discussed in this paper stores all the commonly used suffix and prefix for all clinical root words in its database. This approach prefers time and accuracy to memory space. We confer some of the rules used to remove suffix and prefix from the clinical words to get the root word. Advantage of this approach is that the user will get the precise results. As sometimes suffix trimming approach in active stemmer provide possible root can result in some extra and indifferent result also. Therefore, this approach is suggested at least for the clinical language in which the number of possible inflections for a word is not infinite.

References

[1] Robert Krovetz. Viewing morphology as an inference process. *In proceedings of the 16th International conference on research and Development in Information Retrieval,* pages 191-202, 1993

[2] G. Salton, A. Wong, and C. S. Yang, "*A vector space model for automatic indexing*", Communications. ACM, vol. 18, no. 11, pp. 613-620, Nov.1975.

[3] W. Kraaij and R. Pohlmann, "*Viewing stemming as recall enhancement*," in Proc. 17th ACM SIGIR Conference., Zurich, Switzerland, Aug. 1996, pp. 40-48.

[4] D.Harman, " *How effective is suffixing?*" J. Amer. Soc. Information Science., Vol. 42, no. 1, pp. 7-15.

[5] J. Xu and W. B. Croft, " *Corpus-based stemming using coocurrence of word variants*", ACM Transactions on Information Systems, vol. 16, no. 1, pp. 61-81,1998.

[6] J. B. Lovins, *"Development of a stemming algorithm,"* Mechanical Translation and Computer Linguistic., vol.11, no.1/2, pp. 22-31, 1968.

[7] J. L. Dawson, *" Suffix removal for word conflation,"* Bulletin of the Association for Literary and Linguistic Computing., vol. 2, no. 3, pp. 33-46, 1974.

[8] M. F. Porter, " *An Algorithm for suffix stripping*," Program, vol. 14, no. 3, pp. 130-137, 1980.

[9] C. D. paice, " *Another stemmer*," SIGIR Forum, vol. 24, no. 3, pp. 56-61, 1990.

[10] R. Krovetz, "*Viewing morphology as an inference process*," in Proceedings. 16th ACM SIGIR Conference., Pittsburgh, PA, 1993, pp. 191-202.

[11] M. Kantrowitz, B. Mohit, and V. Mittal, "*Stemming and its effects on TFIDF ranking*," in Proceedings. 23rd Annual for SIGIR Conference. Athens, Greece, 2000, pp. 357-359.

[12] T. Gustad and G. Bouma, "Accurate *stemming of Dutch for text classification*," Language and Computers., vol. 45, no. 1, pp. 104-117, 2002.

[13] M. Bacchin, N. Ferro, and M. Melucci, "*A probabilistic model for stemmer generation*," Information Processing and Management., vol. 41, no. 1, pp. 121-137,2005.

[14] L. D. Baker and A. K. McCallum, "*Distributional clustering of words for text classification*," in Proceedings. 21st ACM SIGIR Conference., Melbourne, Australia, 1998, pp. 96-103.

[15] F. Pereira, N. Tishby, and L. Lee, "*Distributional clustering of English words*," in Proceedings. 31st Annual meeting on Association for Computational Linguistics. 1993, pp. 183-190.

[16] L. Lee, "*Measures of distributional similarity*," in Proceedings. 37th Annual Meeting on Association for Computational Linguistics. 1999, pp. 25-32.