

Boosting Text Classification through Stemming of Composite Words

Marenglen Biba¹ and Eva Gjati²

Abstract. Text mining is a knowledge intensive process with the main purpose of effectively and efficiently processing large amounts of unstructured data. Due to the rapidly growing amount of raw text available there is a strong need for methods that are capable of dealing with this in terms of automatic classification or indexing. In this context, an essential task is the semantic processing of natural language in order to provide a sound input to the text classification or categorization task. One of the important tasks is stemming which is the process of reducing a certain word to its root (or stem). When a text is pre-processed for mining purposes, stemming is applied in order to bring words from their current variation to their original root in order to better process the natural language with subsequent steps. A challenging task is that of stemming composite words which in many languages form a large part of the daily used vocabulary. In this paper we develop a novel rule-based algorithm for stemming composite words and we show through extensive experiments that the text classification accuracy greatly improves by stemming composite words.

1 Introduction

The growing amount of information which comes in different forms and from different topic areas is becoming available almost everywhere: on the web, in a variety of social networks, on corporate or enterprise Intranets and other information centric applications. Due to the proliferation of information in the form of raw text, which needs to be stored and processed, text mining has gained an increasing attention in the recent years. Unstructured data is the easiest form of data that can be found in any application scenario. As a result, there has been a huge need and demand to design algorithms and methods capable of effectively processing and mining textual data (Aggarwal and Zhai, 2012).

¹ University of New York in Tirana, Albania
e-mail: marenglenbiba@unyt.edu.al

² University of New York in Tirana and University of Greenwich, UK
e-mail: evagjati@unyt.edu.al

By employing techniques from Data Mining, Natural Language Processing, Machine Learning, Information Extraction (IE), Knowledge Management and Information Retrieval (IR), text mining attempts to solve the problem of information overload. This involves the discovery, extraction and pre-processing of large collections of documents, the storage of the intermediate representations and the techniques to analyze these intermediate representations (Feldman & Sanger, 2007). In this context, text mining largely exploits methodologies and techniques from corpus-based computational linguistics which is responsible for transforming raw, unstructured data into more carefully structured intermediate format.

One of the processing steps in the transformation of raw text into a form acceptable by data mining algorithms, is stemming which is a computational linguistic and normalization procedure that attempts to reduce different grammatical forms of a word with the same root to a common form, by removing the words affixes (prefixes and suffixes) (Lovins, 1968). What is very often observed is that most of the times the morphological variants of words have semantically similar terms (roots) which may differ in their affixes. For example, the words: *computer*, *computing*, *computation*, *computational* have all the same root *comput*. The idea is that before retrieving information from documents, stemming techniques are applied on the target data with the aim of reducing the data set and increasing the IR performance and effectiveness. Therefore the total number of distinct terms in a query or a document is reduced, which in turn will also reduce the time of processing of the final output.

Considerable effort has been dedicated through the years to stemmers for different languages such as: English, German, Italian, Swedish, etc, with the main purpose on text pre-processing by stripping away the word affixes to its root (stem) form. Unfortunately little research has been done for other languages and for the task of composite words. Albanian is a language that has not been much researched from the point of view of computational linguistics. The first attempt has only been made in (Sadiku, 2011) where the author introduced the first stemmer for Albanian and performed some experiments on text classification showing good performance of several classifiers on the stemmed documents. However, this work does not handle composite words which is a limitation due to the high number of this kind of words in Albanian.

In this paper we develop a stemming algorithm for handling composite words in stemming. The algorithm is based on a set of rules that are generated through a thorough analysis of the Albanian language morphology and structure. The algorithm is used in text classification tasks and we show that the classifier performs much better after composite words are stemmed, compared with the case of not using stemming. We believe that the results obtained can be generalized to other languages where the composite words are common.

The paper is organized as follows: Section 2 presents stemming as a natural language processing task and some of the most important algorithms for rule-based stemming. Section 3 presents some features of Albanian and Section 4 pre-

sents the developed algorithm. Section 4 presents the experiments and we conclude in Section 5 with some conclusions and directions for future work.

2. Stemming Algorithms

The first studies concerning stemming date back to the 1960s when it was proposed the first stemming algorithm (Lovins, 1968). Some stemming algorithms are known as rule based affix removal language dependent algorithms. In addition, other techniques known as statistical and mixed algorithms are alternative methods to stemming, developed to obviate the performance problems and language dependency difficulties of rule based stemmers.

The simplest stemming algorithm of this group was the Truncate (n) stemmer that truncated words at the n-th symbol, keeping unaffected words with n letters and removing the rest. When the length of the word is small the chances of overstemming errors are increased. Although, this technique is not used in real stemming systems, it can serve for evaluating other algorithms (Paice, 1994). S-stemmer is another approach proposed in (Harman, 1991) which is able to conflate English nouns of plural and singular forms. This algorithm is applied only for words longer than three letters as per below set of rules.

2.1 The Lovin's Stemmer

The Lovin's stemmer was the first developed and well known stemming algorithm (Lovins, 1968). It is a context sensitive algorithm that removes endings, based on the principle of the longest match, and applies a number of contextual rules for preventing the removal of endings that can lead to incorrectly produced stems. This algorithm utilizes an extensive list of 294 endings each associated to one of the 29 contextual conditions of the algorithm which prevent removing the endings in certain circumstances. In addition it utilizes 35 transformation rules, designed to deal with the most frequent exceptions. When a word is presented for stemming an ending that meets a certain condition is found and removed. Next, an appropriate transformation rule is implemented in order to handle the word double consonants and irregular plurals. For example, applying the Lovin's stemmer to the word *nationally* there are two endings that match: *ationally* and *ionally*. The first ending is rejected based on the rule that stem must be at least 3 or more letters long, while the second ending is removed with no restriction producing the stem *nat*.

The advantage of the Lovin's stemmer is that it is very fast, able to handle the removal of double letters of words and the many irregular plurals. The disadvantage of this algorithm is the large amount of time and data, due to the number of special cases and rules that should be developed for each ending, which however will be able to handle only a small number of errors. Furthermore, the Lovin's stemmer often missed to reduce certain endings, due to the technical vocabulary used by the author.

2.2 *The Dawson's Stemmer*

The Dawson's algorithm (Dawson, 1974) can be considered as an improvement and extension of the Lovin's algorithm. Dawson used the same longest-match and single pass nature of Lovin's approach but covers an extensive list of approximately 1200 suffixes. In addition, it utilizes the partial matching technique that match stems that are equal within certain limits, instead of the recoding technique used by Lovin's stemmer that involves a number of transformations based on the letters of a stem.

The aim of this stemmer was to elaborate the rules of the original Lovin's stemmer and to improve any basic error. To achieve this, the first step was to include all plurals and simple suffix combinations that increase the ending list size to five hundred. The second step was to use the completion principle to complete any suffix within the ending list of all variants, flexions and combinations that increase the ending list to 1200 terms (Dawson, 1974). However, this extended list of suffixes presented two main problems: the storage limitation and the time required to test all suffixes with the list of endings. To obviate these problems the suffixes were stored in reversed order indexed by length and by last letter. Due to its complexity and deficiency of a standard reusable implementation the Dawson stemmer did not gain popularity.

2.3 *The Porter's Stemmer*

The Porter's stemmer, firstly proposed in the 1980s, is as of now one of the most popular and widely used suffix removal algorithms (Porter, 1980). Quickly adopted and extended, the Porter's algorithm became the word conflation standard approach for IR and a great inspiration for many later stemmers for English and a broad range of other languages. This stemmer was initially developed with the purpose of stemming texts in English language. Later on, the increased importance of IR systems in the 1990s resulted in an intense interest in conflation methods development that would improve text retrieval and provide a natural model for text processing in different languages.

Porter's algorithm employs a simple design of 60 suffixes, 2 rules responsible for recoding and a single context sensitive rule which decides if a suffix should be cut off or not (Willett, 2006). Implemented in five concise steps, this algorithm proved to be efficient in terms of computation complexity. Despite the problems of mis-stemming and over-stemming errors, which can potentially be reduced by using a comprehensive well structured dictionary, this algorithm continues to have good practical results and great utility in IR systems.

The lack of readily available stemming algorithms for languages other than English and the many incorrect implementations and misinterpretations of the original Porter's stemmer in different studies forced Porter to develop a rigorous framework for defining stemming algorithms, known as Snowball (Porter, 2001).

Snowball is a rigorous system able to define stemming algorithms, where rules are expressed in natural way allowing programmers to develop their own stemmers for a chosen language. This framework includes an improved version of English stemmer together with a series of stemmers for other languages.

There are two main reasons why Porter's stemmer is considered important. First, the utilization of a simple technique to stemming that obviously gives good results in practice and can be implemented in many languages. Second, the tremendous interest of researchers in stemming as a separate research IR topic that this algorithm induced.

3. Albanian Language and Composite Word Formation

The Albanian language is part of the extensive Indo-European language family and thus related to a certain extent to almost all the other European languages. Nevertheless, studies have shown no close historical affinity of Albanian to any of the other language of the Indo-European family forming a distinct and unique language branch, which is the Albanian branch (Agalliu, et al., 2002).

From the morphological and structural point of view, Albanian words are considered as diversified and can be broken down in different groups and subgroups. From the point of view of the number of rooting morphemes, the words can be divided in two large groups:

- Simple words, which are created by a single rooting morpheme such as: *breg, bregore, drejt, drejtoj, i drejtë, punë, etc.*
- Composite words, which are created by two or more rooting morphemes *atdhe, armëpushim, dyvjeçar, juglindor, zemërmirë, etc.*

3.1 Composite Formation of Nouns

Composite words according to the syntactic relation between the parts are divided in two groups: composites with conjunctive relation and composites with subordinate relation (Agalliu, et al., 2002). Most composite nouns are those with subordinate relation where we distinguish the following types of composites:

- Noun + adverbial noun of the actor, formed with the *-(ë)s* suffix
- Noun + adverbial noun of the action, formed with *-je* or *-im* suffixes
- Noun + any other noun
- Pronoun or Numeral + Noun prefix *vetë-*
- Adjective + Noun
- Verb + Noun
- Noun + verb formed with suffix *-je*
- Noun + adverb formed with *bashkë-, drejt-, keq-, mirë- kundër-* prefixes

- Composites formed with nouns + agglutinations

3.2 *Composite Formation of Adjectives*

The composite adjectives are created concatenating two or sometimes three themes in a single word (Agalliu, et al., 2002). Like in composite nouns the composite adjectives consist of the same groups of composites:

1. Composites with conjunctive relation

This first group contains those adjectives which are formed without nodes and are equivalent from syntactic and semantic perspectives. The adjectives of this group neither define each other nor are dependent on each other. However they are mutually complementary to a certain extent to each-other.

2. Composites with subordinate relation

A special feature of this group is the determinant character of one of the composite parts. According to the word formation themes that form the adjectives of this group we distinguish the below subgroups:

- Composite adjectives formed with two nominal themes
- Composite adjectives formed with a noun + adjective
- Composite adjectives formed with two adjective themes
- Composite adjectives formed with an adverbial theme + noun
- Composite adjectives formed with a numeral + adjective
- Composite adjectives formed with a participle + adverb

4. **Language Analysis and Algorithm Design**

Almost all composite splitting algorithms developed for other languages use huge knowledge resources: like monolingual lexicons, monolingual or bilingual corpora and parallel corpora. These approaches rely on the availability of lexicons with compound parts using sometimes lazy learning and mostly language specific techniques.

Taking into consideration the pros and cons of rule based and dictionary based algorithms, as well as available resources for Albanian we decided to develop a rule based algorithm. The algorithm contains a number of rules each of which comprises one or more ‘if’ conditions depending on the various cases of composite formations. The set of rules has been conducted analyzing the linguistic and general characteristics of composite words. Each of the rules is executed one by one in a specified order. If the specified conditions are met the algorithm returns the stem of the word. For developing the algorithm we analyzed some groups of composites with common features and developed rules for each of them.

In the first group there are composites that contain a separating dash between the two word parts such as: *ekonomiko-shoqëror*, *politiko-ekonomik*, etc. These types of composites are very frequent in Albanian.

The second group of composites handled by the algorithm consists in composites formed with prefixes. In Albanian, this group of composites is very productive. A list of all possible prefixes has been conducted and integrated in a set of 60 rules. In addition, in this group of composites we distinguish composites formed with international prefixes such as: *hidro*, *bio*, *aero*, *biblio*, *deci*, *kilo*, etc. These composites are also very frequent in Albanian. A set of 29 such rules has been developed to handle these cases.

Composites formed with numerals are the third group of composites handled by the algorithm. A set of 10 rules has been developed considering the most frequent numeral composites.

The fourth group of composites considers a number of productive nouns which associated with adjectives or verbs form a large number of composites. Such nouns are: *jetë*, *gojë*, *ndihmës*, *zëvendës*, *zemër*, etc. 59 rules have been developed for these cases.

The fifth group of composites considers some of the most frequent encountered second composite parts. This rule is created for handling composites that are randomly created and no common rule exists for their formation such as: *bukëpjekës*, *samarpunues*, *orëndreqës*, *gurskalitës*, etc. In addition, this ensures us that the algorithm will handle as many composites as possible.

The sequences of steps that the stemmer follows are illustrated in Fig. 1. All the other steps after splitting are the same as in the JStem algorithm of (Sadiku, 2011). The full set of the above rules coded in Java is presented in (Gjati, 2013).

5. Experimental Evaluation

In this section we will evaluate the performance of the designed algorithm in text mining tasks with documents in Albanian. The effectiveness of the stemming algorithm will be measured in terms of classification accuracy in a rich collection of documents. We will use different algorithms in order to prove that the performance improvement is not just due to a random coincidence with one algorithm, but can be generalized to the whole learning and text classification problem.

5.1 Experimental Setting

In order to measure the performance of the stemmer under the text mining perspective we need to perform text mining experiments. The experiments have been performed in Weka³, an open source program that is widely used in academia and industry. Weka offers a number of tools for text pre-processing, classification, clustering, regression, etc and provides an integrated framework for machine learning and mining (Witten et al., 2011).

³ Weka can downloaded from <http://www.cs.waikato.ac.nz/ml/weka/>.

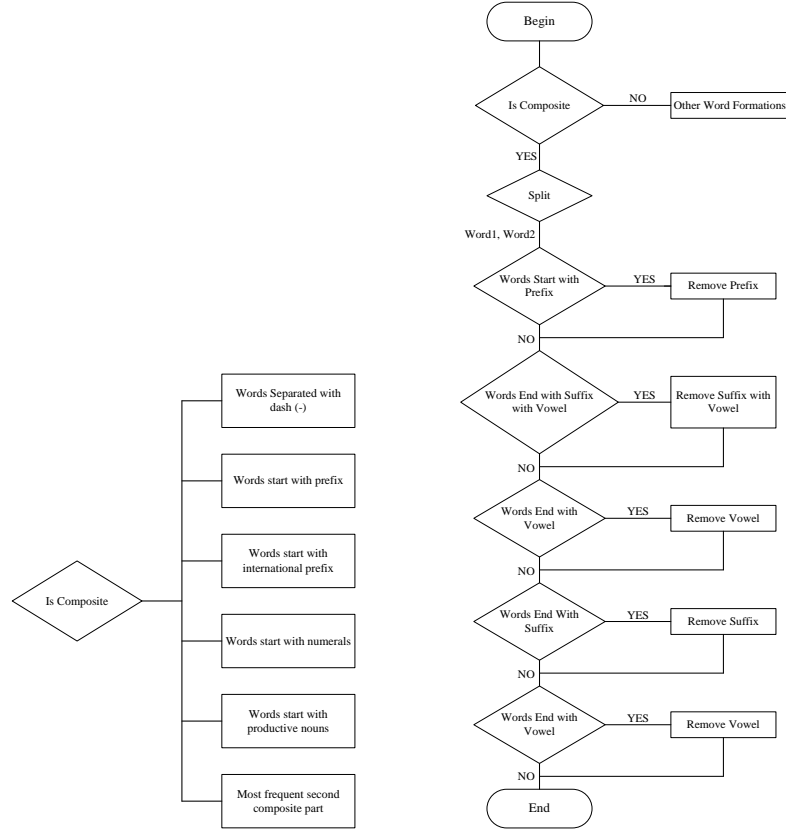


Fig. 1: Groups of composites (left) and algorithm steps (right)

In our experiments we used a corpus developed in (Taullaj, 2012), a previous work done for comparing text mining algorithms for Albanian. The corpora consist of eight different areas of study (Biology, Chemistry, Culture, Curiosities, History, Economy, Literary and Sport) containing 40 documents each. Each document of the corpora is first stemmed by our algorithm and then converted by Weka with the StringToWordVector filter that transforms text files into vectors that are stored in input files in .arff format. These files are then used as training and testing input for the text classification task. The data mining algorithms chosen in these tests are the Nearest Neighbour (IBK in Weka), Support Vector Machines and Naïve Bayes.

5.2 Experimental Results

The experiments were performed between two or more datasets comparing the accuracy of the classifying algorithms in related and unrelated fields. We performed

in total 10 experiments with stemmed and not-stemmed files in order to see if stemming will help in document classification. In each experiment, we used 10-fold cross-validation.

The experiments and the relative classes are as follows:

- 1 Biology and Chemistry
- 2 Biology and History
- 3 Chemistry and Literary
- 4 History and Literary
- 5 Sport and Culture
- 6 Biology, Chemistry and Economy
- 7 Biology, Chemistry, History and Literary
- 8 Culture, Curiosities, History and Literary
- 9 6 Classes: Biology, Chemistry, History, Literary, Economy and Culture
- 10 8 Classes: Biology, Chemistry, History, Literary, Economy, Culture, Curiosities, Sport

Exp.	Stemmed Documents			Terms	Not Stemmed Documents			Terms
	KNN	SVM	NB		KNN	SVM	NB	
1	85%	95%	96.25%	1897	57.5%	68.75%	78.75%	1638
2	90%	93.75%	91.25%	1744	88.75%	96.25%	91.25%	1770
3	96.25%	100%	98.75%	2004	80%	96.25%	95%	1811
4	93.75%	100%	96.25%	1804	87.50%	91.25%	88.75%	1604
5	55%	87.5%	91.25%	2587	55%	86.25%	91.25%	2579
6	81.66%	95%	97.50%	2569	44.16%	79.1667%	89.8333%	2344
7	75%	93.75%	91.87%	3185	55%	76.25%	76.875%	2791
8	73.12%	89.375%	91.87%	4179	55.62%	86.87%	88.75%	3964
9	69.58%	93.33%	94.58%	5201	51.66%	83.33%	84.58%	4835
10	52.50%	88.75%	91.25%	7117	39.37%	80.31%	83.43%	6739

Table 1. Experimental results for the ten experiments

As we can see from the Table 1, the classifier accuracy is significantly higher in the case of using stemmed documents. There is not even one case where the classifier performs better with the not stemmed documents. This is an indication that the stemming algorithm, including the composite word split, leads to significant improvements in the text classification task. In addition, the fact that all the algorithms show a growth of performance in the case of stemmed documents indicates that stemming of composite words is a task that helps the overall learning process and not just particular algorithms.

6. Conclusion

Text mining is a knowledge intensive process that requires techniques from natural language processing in order to provide an appropriate transformation of raw

text into input for machine learning algorithms. In this paper we present a novel stemming algorithm that is able to split composite words and then stem these. We show through extensive experiments on text classification tasks with several machine learning algorithms that the classifiers perform significantly better in the case of stemmed documents compared with the not stemmed case. As future work we intend to develop machine learning models that can learn from corpora how to stem composite words. This however requires large labeled corpora in order to train the model.

References

1. Agalliu, F., Angoni, E., Demiraj, S., Dhrimo, A., Hysa, E., Lafe, E., (2002). *Gramatika e Gjuhes Shqipe (Vol. 1)*. Akademia e Shkencave, ISBN: 99927-761-6-1, Tirane 2002.
2. Aggarwal, C. C., & Zhai, C. X. (2012). *Mining Text Data*. London: Springer, 2012.
3. Dawson, J. (1974). Suffix removal and word conflation. *Bulletin of the Association for Literary and Linguistic Computing*, Vol 2 (3), pp. 33 – 47, 1974.
4. Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook*. New York: Cambridge University Press CB2 8RU, UK, 2007.
5. Gjati, E. Handling Composite Words in Stemming Albanian in a Text Classification Approach. Master Thesis. University of New York Tirana, 2013.
6. Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, 42: 7–15, 1991.
7. Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, Vol 1, nos 1 & 2, 1968.
8. Paice, C. D. (1990). Another stemmer. *ACM SIGIR Forum*, Vol 24(3), pp. 56 – 61, 1990.
9. Paice, C. D. (1994). An evaluation method for stemming algorithms. *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Vol 17, pp. 42 - 50. UK.
10. Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program Electronic Library and Information Systems* Vol 14 (3), pp 130-137, 1980.
11. Porter, M. F. (2001, October). Retrieved November 2012, from Snowball: A language for stemming algorithms: URL: <http://snowball.tartarus.org/texts/introduction.html>
12. Porter, M. F. (2006). Stemming algorithms for various European languages. Retrieved from <http://snowball.tartarus.org/texts/stemmersoverview.html>
13. Sadiku, J. (2011). *A Novel Stemming Algorithm for Albanian text in a Data Mining Approach for Document Classification*, Master Thesis. Tirana: University of New York Tirana, 2011.
14. Taullaj, L. (2012) Comparative Evaluation of Text Mining Algorithms for Albanian. Master Thesis. University of New York Tirana, 2012.