2015 The 5th International Workshop on Computer Science and Engineering
doi: 10.18178/wcse.2015.04.044

A New Multilingual Stemmer Based on the Extraction of the Root

Said Gadri[1] and Abdelouahab Moussaoui[2]

[1]Department of ICST, University of M'sila, M'sila, 28000, Algeria

[2]Department of Computer Sciences, University Farhat Abbes of Setif, Setif, 19000, Algeria

Keywords: Root extraction, stemming, information retrieval, bigrams technique, text mining, machine learning, natural language processing.

Abstract. Stemming is a technique used to reduce inflected and derived words to their basic forms (stem or root). It is a very important step of pre-processing in text mining, and generally used in many areas of research such as: Natural language Processing NLP, Text Categorization TC, Text Summarizing TS, Information Retrieval IR, and other tasks in text mining. Stemming is frequently useful in text categorization to reduce the size of terms vocabulary, and in information retrieval to improve the search effectiveness and then gives us relevant results.

In this paper, we propose a new multilingual stemmer based on the extraction of word root and in which we use the technique of n-grams. We validated our stemmer on three languages which are: Arabic, French and English.

Introduction

Text categorization process consists of assigning a set of texts to a set of predefined categories. For this purpose, we use generally many algorithms known in machine learning such as: K-NN, SVM, RBF, NB, etc. During the process of TC, the document must pass through a series of steps: removing punctuation and stop words, representing each document with a vector of terms, calculation of terms frequencies TF, and inverse document frequencies TF-IDF. One of the problems which we can meet is the big size of vectors used in the representation of documents, especially when we work on a big corpus of texts like "Reuters". To solve such problem, several statistical methods are used to select some relevant terms in order to use them in the entry of learning algorithms. These methods allows us to reduce the dimension of the vector space representing the different documents in one hand, in the other hand, it permits to improve the quality of categorization process. Among these methods we can note: the mutual information MI, the information gain IG, and Khi2 law. Another method that seems very effective especially for Arabic TC is the selection of relevant terms by stemming. Stemming is a technique in which we replace dozens of terms (words) which occur in different documents and semantically close by their basic forms (stems or roots) in order to reduce the dimension of terms vector and thus increase the quality of obtained categorization. Several stemmers are developed for various languages as English, French, German and Arabic, but each one has its own advantages as well as limitations. Most of the stemming algorithms used in this field are language dependent [1]. So, it is important to develop a new stemmer which is language independent. In this way, we propose in our work a new multilingual stemmer based on the extraction of the word root, as well as the use of n-grams technique. The proposed stemmer was tested on three languages which are: Arabic, French, and English and gave promising results. The paper is organized as follows: the second section presents some related works, so we review some papers that treat the problem of stemming and the used approaches. In the third section we introduce our new stemming algorithm. The fourth section presents the experiments done to test our new stemmer and the obtained results. In the last section we conclude our work by summarizing our work and giving some ideas to improve it in the future.

Related Work

Stemming algorithms can be classified in three groups: truncating methods, statistical methods, and mixed methods [2]. Each of these groups has a typical way of finding the stems of the word variants. The first group is related to removing the affixes of a word. This was the first stemmer proposed by Lovins in 1968 [3]. The Lovins stemmer removes the longest suffix from a word. Once the ending is removed, the word is recoded using a different table to convert these stems into valid words. The advantages of this algorithm is it is very fast. Drawbacks of the Lovins approach are that it is time and data consuming. Porter Stemmer [4, 5] is until now one of the most popular stemming methods proposed in 1980. It is based on the idea that the suffixes in English are mostly made up of a combination of smaller and simpler suffixes. It has five steps, and within each step, many rules are applied until one of them passes the conditions. The resultant stem at the end of the fifth step is returned. The Paice/Husk stemmer is an iterative algorithm with one table containing about 120 rules indexed by the last letter of a suffix [6]. On each iteration, it tries to find an applicable rule by the last character of the word. Each rule specifies either a deletion or replacement of an ending. The advantage is its simple form, and every iteration taking care of both deletion and replacement as per the rule applied. The disadvantage is it is a very heavy algorithm and over stemming may occur. Dawson Stemmer [7] is an extension of the Lovins approach except that it covers a list of about 1200 suffixes. The advantage is that it covers more suffixes than Lovins and is fast in execution. The disadvantage is it is very complex and lacks a standard reusable implementation. The second group is called statistical methods; it contains stemmers which are based on statistical techniques. Most of the methods remove the affixes but after implementing some statistical procedures. In this group we can find the following stemmers: N-Grams Stemmer [6], [8]: it is a language independent stemmer, the main idea behind this approach is that, similar words will have a high proportion of n-grams in common. The advantage that it is language independent and hence very useful in many applications. The disadvantage is it requires a significant amount of memory and storage for creating and storing the n-grams. The HMM Stemmer was proposed by Melucci and Orio [9] and based on the concept of the Hidden Markov Model (HMMs) which are finite-state automata. At each transition, the new state emits a symbol with a given probability. This method does not need a prior linguistic knowledge, and the most probable path is found using the Viterbi coding in the automata graph. The third group is called the mixed methods which contain: The Inflectional and Derivational Methods which involves both the inflectional and the derivational morphology analysis. Here, the corpus should be very large to develop these types of stemmers. In case of inflectional, the word variants are related to the language specific syntactic variations like plural, gender, case, etc. Whereas, in derivational the word variants are related to the part-of-speech (POS) of a sentence where the word occurs. Krovetz Stemmer (KSTEM) was presented in 1993 by Robert Krovetz [10] and it is a linguistic lexical validation stemmer. It effectively removes inflectional suffixes in three steps. Since this stemmer does not find the stems for all word variants, it can be used as a pre-stemmer before applying a stemming algorithm. This would increase the speed and effectiveness of the main stemmer. Xerox Inflectional and Derivational Analyzer; The linguistics groups at Xerox have developed a lexical database for English and some other languages also which can analyse and generate inflectional and derivational morphology. The inflectional database reduces each surface word to the form which can be found in the dictionary, as follows [11]: nouns singular (e.g. children child), verbs infinitive (e.g. understood, understand), etc. The advantages of this stemmer are that it works well with a large document and removes the prefixes, all stems are valid words. The disadvantage is that the output depends on the lexical database which may not be exhaustive. So, it cannot correctly stem words which are not part of the lexicon. The Corpus Based Stemmer was proposed by Xu and Croft [8]. It refers to automatic modification of conflation (classes – words) that have resulted in a common stem, to suit the characteristics of a given text corpus using statistical methods. The basic hypothesis is that word forms that should be conflated for a given corpus will co-occur in documents from that corpus. Using this concept some of the over stemming or under stemming drawbacks are resolved. The advantage of this method is it can potentially avoid making conflations that are not appropriate for a given corpus and the result is an actual word and not an incomplete stem. The disadvantage is that you need to develop the statistical

measure for every corpus separately and the processing time increases as in the first step, and stemming algorithms are first used before using this method. Context Sensitive Stemmer is done using statistical modelling on the query side. It was proposed by Funchun Peng et. al [12]. Basically for the words of the input query, the morphological variants which would be useful for the search are predicted before the query is submitted to the search engine. This reduces the number of bad expansions, which in turn reduces the cost of additional computation and improves the precision at the same time. The advantage of this stemmer is it improves selective word expansion on the query side and conservative word occurrence matching on the document side. The disadvantage is the processing time and the complex nature of the stemmer. Many stemming algorithms based on the previous approaches have been developed for a wide range of languages including English [13], Latin [11], German and Italian [14], French [15], Chinese [16]. For Arabic Language, there are three different Stemming approaches: the root-based approach [17-19]; the light stemmer approach [20, 21]; and the statistical stemmer approach (N-Grams) [22-23]. Yet no a complete stemmer for this language is available.

The Proposed Stemmer

In our work, we proposed a new multilingual stemmer, in which we use also the n-grams technique to extract the root of a word belonging to one of the following languages: Arabic, French, and English. For this purpose, we proceed according to the following steps:

Step 1: we segment the word for which we want to find the root, and all the roots of the predefined list into bigrams (2-grams).

For example if we have the words: "يذهبون" in Arabic, "calculateur" in French, and "bellicism" in English, and the three lists of roots in the above languages as follows: (فتح ، خرج ، ذهب ، وهب), (assist, calcul, compt, conclu), (sciss, bell, dict, tele), we proceed the segmentation step as indicated in the following table:

Table 1: An Example Describing the Segmentation Step (Step 1)

Language	Word (W)/Bigrams	List of roots (Ri)/Bigrams
Arabic	يذهبون	R_1 = "فتح" ➔ (ح فت ،)
	يذ، ذه ، هب ، بو ، ون	R_2 = "خرج"➔ (خر ، رج)
		R_3 = "ذهب" ➔ (ذه ، هب)
		R_4 = "وهب" ➔ (وه ، هب)
French	calculateur	R_1 = "assist" ➔ (as, ss, si, is, st)
	ca al lc cu ul la at te eu ur	R_2 = "calcul" ➔ (ca, al, lc, cu, ul)
		R_3 = "compt" ➔ (co, om, mp, pt)
		R_4 = "conclu" ➔ (co, on, nc, cl, lu)
English	bellicism	R_1 = "sciss" ➔ (sc, ci, is, ss)
	be el ll li ic ci is sm	R_2 = "bell" ➔ (be, el, ll)
		R_3 = "dict" ➔ (di, ic, ct)
		R_4 = "tele" ➔ (te, el, le)

Step 2: We calculate for each word the following parameters:

N_w: The number of bigrams in the word W

N_{R_i}: The number of bigrams in the root R_i

N_{wR_i}: The number of common bigrams between the word W and the root R_i

$N_{w\bar{R}_i}$: The number of bigrams belonging to the word W and do not belong to the root R_i ($N_{w\bar{R}_i} = N_w - N_{wR_i}$)

$N_{R_i\bar{W}}$: The number of bigrams belonging to the root R_i and do not belong to the word W ($N_{R_i\bar{W}} = N_{R_i} - N_{wR_i}$).

For the previous example we have:

Step 3: We take only the roots having at least one common bigram with the word W ($N_{wR_i} \geq 1$) as candidate roots among the list of all roots in order to reduce the calculation time. In our previous example, we have:

Table 2: Calculation of Word Parameters (Step 2)

Word (W)	N_w	Associated roots R_i	N_{R_i}	N_{wR_i}	$N_{w\bar{R}_i}$	$N_{R_i\bar{W}}$
يذهبون	05	R_1 = "فتح", R_2 = "خرج"	2, 2	0, 0	5, 5	2, 2
		R_3 = "ذهب", R_4 = "وهب"	2, 2	2, 1	3, 4	0, 1
calculateur	10	R_1 = "assist, R_2 = "calcul"	5, 5	0, 5	10, 5	5, 0
		R_3 = "compt", R_4 = conclu"	4, 5	0, 0	10, 10	4, 5
bellicism	08	R_1= "sciss, R_2 = "bell"	4, 3	2, 3	7, 6	2, 0
		R_3 = "dict", R_4 = "tele"	3, 3	1, 1	8, 8	2, 2

Table 3: Selection of Candidate Roots (Step 3)

Word (W)	N_w	Associated roots R_i	N_{R_i}	N_{wR_i}	$N_{w\bar{R}_i}$	$N_{R_i\bar{W}}$
يذهبون	05	R_3 = "ذهب", R_4 = "وهب"	2, 2	2, 1	3, 4	0, 1
calculateur	10	R_2 = "calcul"	5	5	5	0
bellicism	08	R_1= "sciss, R_2 = "bell"	4, 3	2, 3	7, 6	2, 0
		R_3 = "dict", R_4 = "tele"	3, 3	1, 1	8, 8	2, 2

Step 4: We calculate the distance $D(W, R_i)$ between the word W and each candidate root R_i according to the following equation :

$$D(W, R_i) = (N_{w\bar{R}_i} + N_{R_i\bar{w}})/(N_w + N_{R_i}) \quad (1)$$

For the previous example we obtain:

Table 4: Calculation of the distance between the word and each candidate Root

Word (W)	N_w	Associated roots R_i	N_{R_i}	N_{wR_i}	$N_{w\bar{R}_i}$	$N_{R_i\bar{W}}$	$D(W,R_i)$
يذهبون	05	R_3 = "ذهب", R_4 = "وهب"	2, 2	2, 1	3, 4	0, 1	**0.42**, 0.71
calculateur	10	R_2 = "calcul"	5	5	5	0	**0.33**
bellicism	08	R_1= "sciss, R_2 = "bell"	4, 3	2, 3	7, 6	2, 0	0.69, **0.5**
		R_3 = "dict", R_4 = "tele"	3, 3	1, 1	8, 8	2, 2	0.83, 0.83

Step5: In the last step, we assign the root that has the lowest value of distance $D(W, R_i)$ among the candidate roots to the word W. it is the required root.

In our example, the roots of the given words are:

Table 5: Extraction of the word root (Step 5)

Word *(W)*	Extracted root *(R)*	Effective root
يذهبون	ذهب	ذهب
calculateur	calcul	calcul
bellicism	bell	bell

Finally, we note that our new algorithm has the following advantages:

1. It is language independent that means it is applicative for any language
2. Does not require the removal of affixes.
3. Works for any word whatever the type of the root. (e.g., trilateral roots, quadrilateral, quinquelateral, and hexalateral roots in Arabic).
4. Valid for strong roots and vocalic roots, which pose generally problems in Arabic during their derivation, because the complete change of their forms.
5. Does not use any morphological rule but only calculations of distances.
6. Very practical stemmer and easy to implement on machine.

Experimentation and Obtained Results

To validate our proposed stemmer, we have used the following data set:

Table 6: Data Set Used in Experimentation

Corpus	Language	Size of derived words' file	Size of the roots' file	Size of the golden roots' file
Small corpus	Arabic	50	25	50

	French	44	36	44
	English	92	56	92
Middle	Arabic	270	140	270
corpus	French	180	220	180
	English	346	165	346
Large corpus	Arabic	750	450	750
	French	680	585	680
	English	720	545	720

Table 7: Extraction of Some Words Roots Using the New Stemmer

Word (*W*)	Nearest Roots R_i	Nb.Common	Distance Values	Extracted	Correct
		Arabic			
يتعلمون	كلم ، علم ، علم ، علج	1، 1 ،3 ، 2	0.77 , 0.77 , **0.4**, 0.6	علم	علم
سنستدرجهم	درج	2	**0.6**	درج	درج
		French			
Apprentissage	Apprend, assembl, assist, associ, assur, autoris, chang, comprend	4, 1, 2, 1, 2,1, 1,2	**0.55**, 0.9, 0.78, 0.89, 0.77, 0.9, 0.88, 0.78	apprend	apprend
connaissance	Assist, avanc, command, commenc, concev, conclu, conna î	2, 3, 2, 3, 4, 3, 4	0.77, 0.64, 0.78, 0.68, **0.57**, 0.66, 0.57	concev	conna î
		English			
transactions	action, extra, intra, dict	5, 2, 2, 1	**0.28**, 0.69, , 0.69,	action	action
geopolitics	Action, geo, poly	1, 2, 2	0.86, **0.66**, 0.69	geo	geo

Table 8: Obtained Results When Extracting the Words Roots

Corpus	Language	Nb.roots	Nb.Words	Corrt	Wrong	Suc_rate	Err_rate
Small	Arabic	25	50	47	03	94,00	06,00
corpus	French	36	44	41	03	93,18	06,82
	English	56	92	85	07	92,39	07,61
Middle	Arabic	140	270	198	72	73,33	26,67
corpus	French	220	180	152	28	84,44	15,56
	English	165	346	309	37	89,30	10,70
Large	Arabic	450	750	519	231	69,20	30,80
corpus	French	585	680	539	141	79,26	20,74
	English	545	720	623	97	86,52	13,48

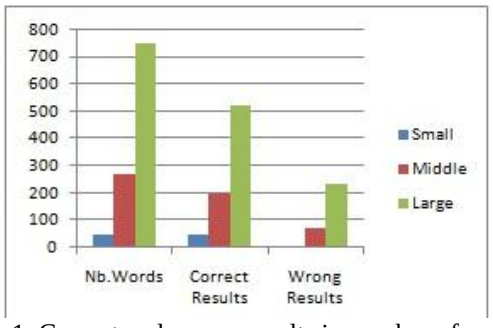

Fig. 1. Correct and wrong results in number of words (Arabic)

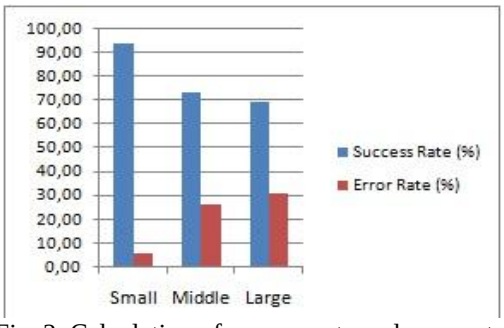

Fig. 2. Calculation of success rate and error rate (Arabic).

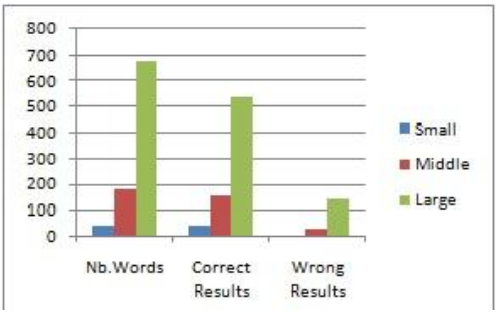

Fig. 3. Correct and wrong results in number of words (French).

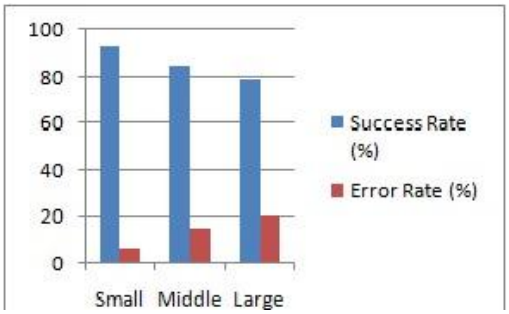

Fig. 4. Calculation of success rate and error rate (French).

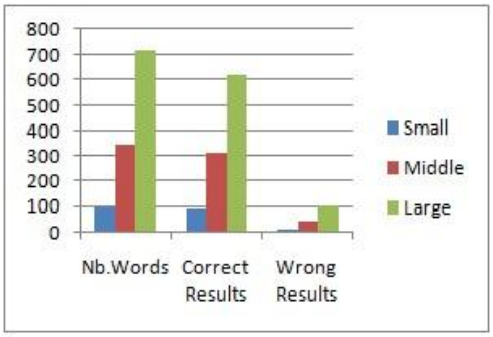

Fig. 5. Correct and wrong results in number of words (English).

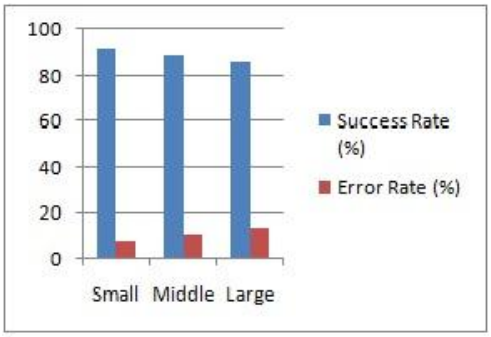

Fig. 6. Calculation of success rate and error rate (English).

Conclusion and Perspectives

In this paper we have studied the problem of stemming and its positive influence on TC, IR, and other areas of NLP and text mining especially for the reduce of terms vocabulary (TC), and the increase of the effectiveness in search engines (IR). We exposed the most known approaches and techniques in the field, notably: truncating methods, statistical methods, and mixed methods. For each one, we gave their advantages and their weaknesses. In the light of the studied approaches, we proposed a new multilingual stemmer, which is language independent, based on n-grams technique, and does not require prior linguistic knowledge related to the above studied languages. Our stemmer is able to find all the types of roots, especially for Arabic, which is a very rich language, having a difficult structure and a complex morphology. The obtained success rates of the root extraction for the previous three languages are very promising and can be improved in future works.

References

[1] Hadni M, Ouatik S.A. & Lachkar, A. Effective Arabic stemmer based hybrid approach for Arabic TC. *Int.J.of.DM & KM Process (IJDKP)*,3(4), 2013

[2] Jivani A. G. A Comparative Study of Stemming Algorithms, *Int. J. Comp. Tech. Appl IJCTA*, 2 (6), pp.1930-1938, 2011.

[3] Lovins J. B. Development of a stemming algorithm, *Mechanical Translation and Computer Linguistic*, 1(2), pp. 22-31, 1968.

[4] Porter M.F. An algorithm for suffix stripping. *Program*, 14, 130-137, 1980.

[5] Porter M.F. Snowball: A language for stemming algorithms, 2001.

[6] Chris P. D. Another stemmer. ACM SIGIR Forum, 24(3). pp. 56-61, 1990.

[7] Dawson J. Suffix removal and word conflation, *Bulletin of the Association for Literary and Linguistic Computing*, 2(3), pp.33–47, 1974.

[8] Jinxi X & Croft Bruce W. Corpus-based stemming using co-occurrence of word variants. *ACM Trans. Info. Systems*, 16(1), pp. 61-81, 1998.

[9] Melucci, M., & Orio, N., A novel method for stemmer generation based on HMM models. Proc.of 11th.inter.conf.on Inf&KM., pp.131-138, 2003.

[10] Krovetz R. Viewing morphology as an inference process. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. pp.191-202, 1993.

[11] Hull D. A., a& Grefenstette,. A detailed analysis of English Stemming Algorithms, *XEROX Technical Report* available on: http://www.xrce.xerox.

[12] Funchun, P., Nawaaz, A., Xin, L., & Yumao, L., Context sensitive stemming for web search. In Proc.of the 30th Inter.ACM SIGIR conference on Research and development in information retrieval. pp. 639-646, 2007.

[13] Greengrass, M., Robertson, A. M., Robyn, S. &Willett, P. Processing morphological variants in searches of Latin text. *Inf.Res.News*, 6(4), 1996.

[14] Monz C. & de Rijke, M. Shallow morphological analysis in monolingual information retrieval for German and Italian. In Cross-language information retrieval and evaluation: Proceedings of the CLEF 2001 workshop, C. Peters, Ed.: Springer Verlag, 2001.

[15] Moulinier, I., McCulloh, A., and Lund, E. West group at CLEF 2000: Non-English monolingual retrieval. In Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop, C. Peters, Ed.: Springer Verlag, pp. 176-187, 2001.

[16] A.H.Tan & P.Yu., A Comparative Study on Chinese TC Methods, PRICAI 2000 Workshop on Text and Web Ming, Melbourne, pp.24-35,August 2000.

[17] Khoja S. *Stemming Arabic Text*. Lancaster, U.K., Computing Department, Lancaster University. 1999.

[18] Larkey, L., & M. E. Connell., Arabic information retrieval at UMass in TREC-10". Proceedings of TREC 2001, Gaithersburg: NIST. 2001.

[19] Fatma A.H & Keith E. Emmert. Rule-based Approach for Arabic Root Extraction: New Rules to Directly Extract Roots of Arabic Words, *Journal of Computing and Information Technology – CIT*, 22(1), pp. 57–68, 2014.

[20] Al-omari, A., Abuata, B., Al-kabi, M., Building and Benchmarking New Heavy/Light Arabic Stemmer. The 4th.Inter.conf.on.Info&Com.sys, (2013).

[21] Al-shalabi, R., Kanaan, G., Al-Serhan, H.: New Approach for Extracting Arabic Roots. Proceedings of the International ArabConference on Information Technology (ACIT'2003), Alexandria, Egypt, pp. 42–59, 2003.

[22] Hmeidi, I.I., Al-Shalabi, R., Al-Taani, A.T., Najadat, H., and Al-Hazaimeh, S.A. A novel approach to the extraction of roots from Arabic words using bigrams. *J. Am. Soc. Inform. Sci.Technol*, 61, pp.583-591, 2010.

[23] Yousef, N., Aymen, A.E., Ashraf, O., Hayel, K. An Improved Arabic Word's Roots Extraction Method Using N-grams, *J.Sci.Comp*, 10(4), 2014.