Bon: First Persian Stemmer

M. Tashakori
Tashakori@noavar.com

M. Meybodi
Meybodi@ce.aku.ac.ir

F. Oroumchian
foroumchian@yahoo.com

Abstract:

Stemmers are softwares that find syntactic` roots of the words. They play an important role in natural language processing and other fields such as information retrieval (IR). In IR using stemmed words instead of the original words, could increase as much as 15 percent to the overall performance. In this paper, we report on the development of the first Persian stemmer (Bon). Bon is tested on a collection of Persian texts in the domain of computer science. In our experiments, the recall has been improved by 40 percent.

Keywords:

Data Mining, Data and Knowledge Engineering, Artificial Intelligence, Persian Stemmer

1. Introduction

In Natural languages, we can find limited words that are syntactic' roots of the other words. In an Indo-European language like Persian, a typical word contains a stem (root) which refers to central idea or meaning, and certain affixes have been added to this stem to modify the meaning and/or fit the word for its syntactic role.

Stemming is a widely used method of word standardization designed to allow the matching of morphologically related terms. If, for example, a searcher enters the term *stemming* as part of a query, it is likely that he or she will also be interested in such variants as *stemmed* and *stem*. Stemmers are softwares that extract stems of word automatically [1].

In natural language processing and other fields such as information retrieval (IR), stemmers play an important role. In IR using stemmed words instead of the original words, could increase the level of the exhaustivity of indexing, and could contribute as much as 15 percent to increasing overall performance. Also stemming reduce the size of indexing files. Since a single stem typically corresponds to several full terms, by storing stems instead of terms, compression factors of over 50 percent can be achieved. Thus in this paper we report on the development of the first Persian stemmer which is called Bon. In next section stemming algorithm type of Bon will be described. Then in section 3 we will study some of the properties of Persian words. Section 4 present Bon algorithm and section 5 describes experiments. Finally last section is allocated to conclusions.

2. Persian Words

Persian is an Indo-European language. So in this language there are few stems, and other words are constructed by adding prefixes and suffixes to stems. Bon is an affix removal stemmer. Affix removal algorithms remove suffixes and/or prefixes from terms leaving a stem. These algorithms sometimes also transform the resultant stem. A simple example of an affix removal stemmer is one that removes the plurals from terms [1]. However, in removing Persian affixes, there are many exceptions.

Persian verbs have inflectional property, because they include person, number, and tense. Therefor Bon has a dictionary of infinitives (and in exceptions, present tense of an infinitive). Moreover, infinitives (verbs) in Persian can be simple, or compound, or phrasal. We can find at least one space between components of compound or phrasal infinitives in Persian. For example, in Persian, we have “قسم خوردن”*(ghasam xordan)* that is equal to “oath”, and “از دست دادن” *(az dast dädan)* that is equal to “lose”. So all components of these infinitives (verbs) should be considered as a whole word. We have considered this problem in our stemmer algorithm.

In Persian plural nouns are made by adding “ان”*(än)* or “ها”*(hä)* to the end of nouns. But if any noun ands in a ”ه”*(eh)*; then before adding “ان”*(än)*, “ه”*(eh)* converts to “گ”*(eg)* as shown in the figure 1(a). Also there are nouns which ending in a “ان”*(än)*, but are not plural like “قهرمان”*(ghahramän)*. Moreover some nouns that are adopted from Arabic language have irregular plural form (“Mokassar”) as shown in the figure 1(b). In addition plural form of some nouns are made by adding Arabic plural signs like “ون”*(un)*, ”ين”*(in)*, and “ات”(at) as shown in the figure 1(c). But if a noun ends in a “ا”*(ä)*,”و”*(u)*, ”ه”*(eh)*, or “ي”*(y)*, instead of adding “ات”*(ät)*, “جات”*(jät)* is added, as shown in the figure 1(d) [2].

خواننده ← خوانندگان (*xänandeh* → *xänandegän*)	كتاب ← كتب (*ketäb* → *kotob*)	روحاني ← روحانيون *ruhäny* → *ruhänyun*	كارخانه ← كارخانجات (*kärxäneh* → *kärxänejat*)
		مؤمن ← مؤمنين *momen* → *momenin*	
		تدارك ← تداركات *tadärok* → *tadärokät*	
(a)	(b)	(c)	(d)

Figure 1.

In Persian a pronoun can add to the end of the noun. But if a noun is ending in a “ا”*(ä)*, or “و”*(u)*; then before adding the pronoun, “ي”*(y)* is added to the end of the noun. Examples of this case are “پا ← پايم”*(pä → päyam)* that is equal to “foot → my foot”, and “چاقو ← چاقويش ”(chäghu→chäghuya) that is equal to “knife → his knife”. Also if a singular pronoun is added to the end of the noun and the noun is ending in a “ه”*(eh)*; then before adding the pronoun, “ا”*(a)* is added to the end of the noun. Example of this case is “خانه ← خانهام”*(xäneh→xäneham)* that is equal to “house → my house”

For stemming words that are adopted from Arabic language, we can enter rules and stems from Arabic language or use a table lookup method for these words. Since there are problems with constructing this table, we chose the first method.

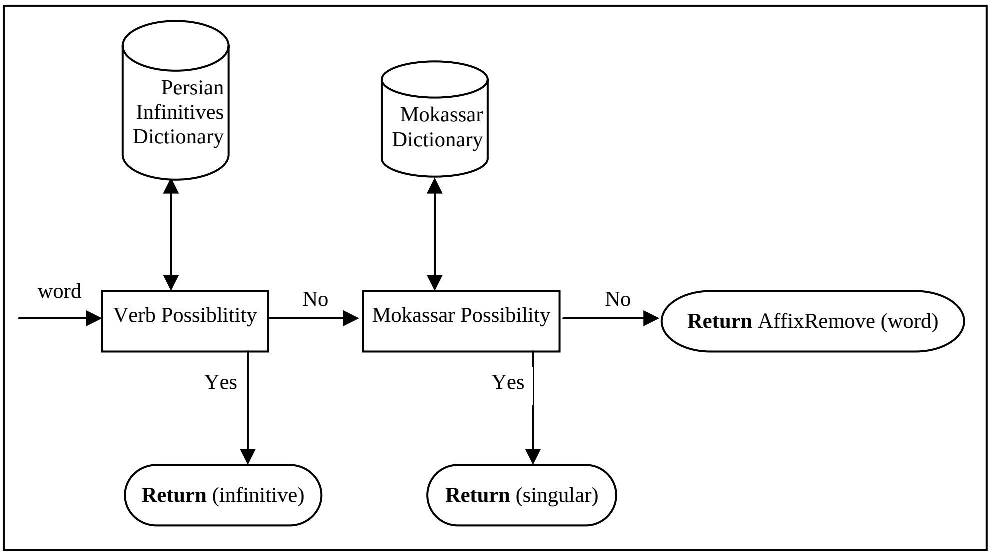
3. Bon Algorithm

Bon such as most stemmers currently in use is an iterative longest match stemmer. An iterative longest match stemmer removes the longest possible string of characters from a word according to a set of rules. This process is repeated until no more characters can be removed. Even after all characters have been removed, stems may not be correct. The word

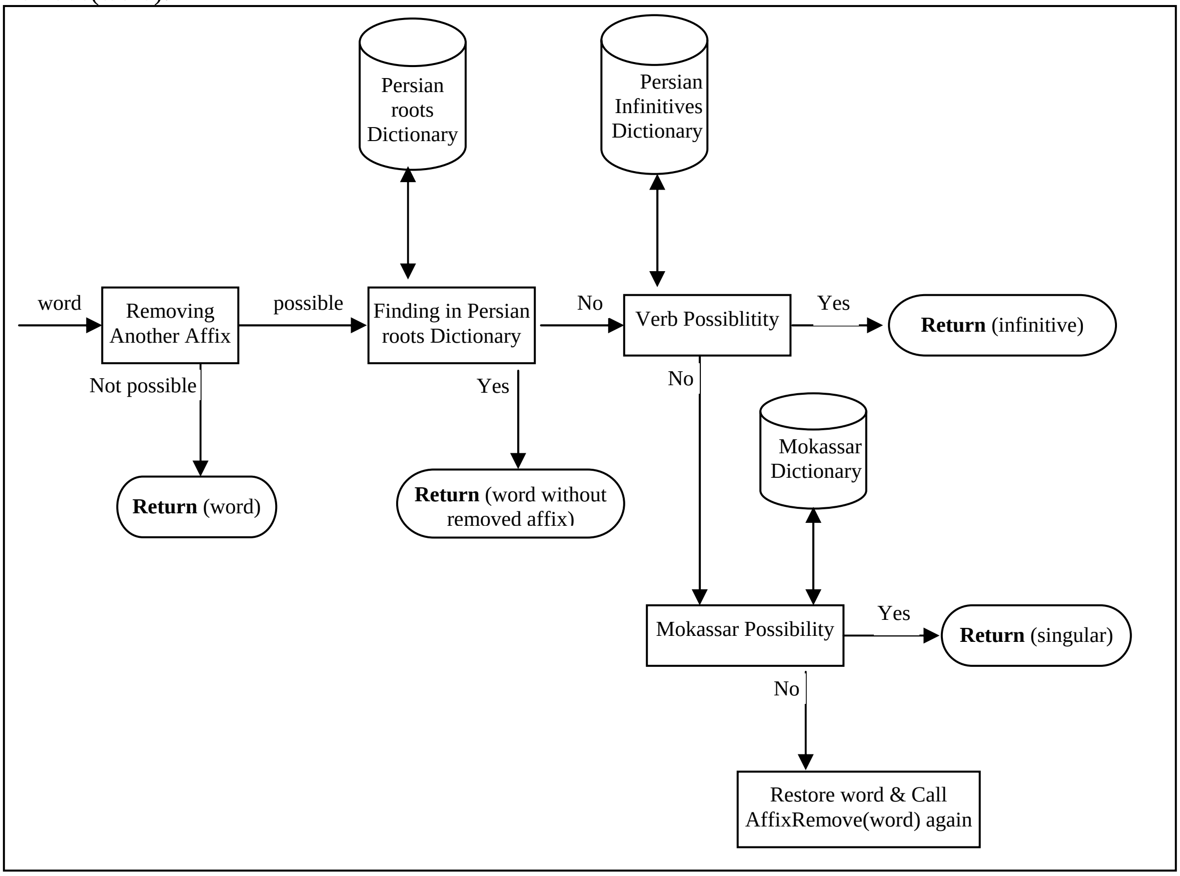
"خانگي"*(xänegy)* that is equal to “house-made”, for example, may have been reduced to the stem "خانگ"*(xäneg)* which will not match "خانه"*(xäneh)* that is equal to “house”. There are two techniques to handle this: recoding or partial matching [1].

Recoding is a context sensitive transformation of the form AxC -> AyC where A and C specify the context of the transformation, x is the input string, and y is the transformed string. In partial matching, only the n initial characters of stems are used in comparing them. Using this approach, we might say that two stems are equivalent if they agree in all but their last character [1]. In Bon recoding technique is being used. For addressing exceptions in Persian words, we have made following components in Bon:

Stem(word):



AffixRemove(word):



A) Stemming rules that are extracted from Persian word constructing rules.
B) A dictionary of Persian infinitives.
C) A dictionary of "Mokassar" words and their singular form
D) A dictionary of insensitive words. Because of difficulties in making this dictionary, we have constructed an experimental dictionary, which has almost 7000 words. For constructing this dictionary, we extracted words from 450 abstracts in our collection. Then by running Bon, words that were derivative of any other word in dictionary were eliminated. Moreover we gradually added or eliminated many words to/from insensitive dictionary.

In Persian letters are attached to one another to form words. So for addressing what words should be attached together and what are separate, we have to consider any word and its followings in the text simultaneously. This consideration is incorporated in Bon.

4. Experiments

Stemmers can be judged on their retrieval effectiveness that usually measured with recall and precision (for definitions of these parameters see [3]), and on their speed, size, and so on. Finally, they can be rated on their compression performance. In this experiment we used recall and precision measurements in an IR system for Bon evaluation.

We have put together a corpus of 450 abstracts of Persian texts in the domain of computer science, which is called PCA (Persian Computer Abstracts). Experiments were performed on this collection using 32 queries.

In document retrieval the needed information is the subset of documents which are deemed to be relevant to the query. So words that cannot possibly be used to identify document content, are eliminated in indexing time. We use a typical stop list in our Persian IR system. This list has 150 Persian high frequency words, and is shown in table 1.

Table 1- typical stop list for a Persian IR system

آنگاه(then)	آنکه(which)	آنچه(whatever)	آنجا(there)	آنان(they)	آن(that)
الان(now)	اگرچه(although)	اگر(if)	است(is)	از(from)	آنها(those)
این(this)	ایشان(they)	اي*	او(he,she)	انجام(do)	اما(but)
باشد(be)	با(with)	اینها(these)	اینگونه(so)	اینست(it's)	اینجا(here)
بنابراین(therefore)	بنابر**	براي(for)	بر(upon,over)	بدون(without)	باید(must,should)
تايي(many)	تا(until)	پس(then)	بین(between)	بیشتر(more)	به(at)
چنان(that)	چگونه(how)	چگونگي(manner)	چرا(why)	توسط(via)	تو(you)
چنین(this)	چندین(many)	چندگانه(multiple)	چندان(so much)	چند(a few)	چنانچه(if)
حتي(even)	چیست(what is)	چیزي(anything)	چیز(thing)	چون(since)	چه(what)
خودش(himself,herself)	خودتان(yourselves)	خودت(yourself)	خود(itself)	خواهیم(will)	خواهد(will)
داراي(having)	داده(given)	خویش(self)	خودمان(ourselves)	خودم(myself)	خودشان(themselves)
دیگر(other)	دو(both)	درباره(about)	در(in)	داشته(having)	دارد(has)
سپس(next)	زیرا(because)	روي(on)	را***	دیگري(another)	دیگران(others)

*part of Persian present perfect verbs for the second person

**part of "بنابر این" (therefore)

شود(be)	شما(you)	شده(becoming)	شد(became)	شايد(may)	شامل(including)
كه(that)	كردن(do)	كرد(do)	كدام(which)	فقط(only)	صورت****
مابين(between)	ما(we)	ليكن(however)	لكن(however)	گرفته(taken)	گرچه(though)
ميتوان(can)	ميباشد(is)	مورد(case)	من(I)	مختلف(several)	مانند(as)
ميكنند(do)	ميكند(do)	ميشوند(become)	ميشود(becomes)	ميدهد(give)	ميتواند(could)
نيست(isn’t)	نيز(again)	نياز(need)	نوع(kind)	نميتوان(cannot)	نظر(seem)
هرچه(whatever)	هر(each)	هايي*****	هاي*****	ها*****	نيستند(aren’t)
همان(same)	هم(too)	هستيم(are)	هستند(are)	هست(is)	هريك(every)
همه(all)	همديگر(each other)	همچون(like)	همچنين(also)	همانند(like)	همانطور(same)
هيچ(never)	هنوز(still,yet)	همين(this)	هميشه(always)	هميشگي(usual)	همواره(ever)
ولي(but)	وگرنه(otherwise)	وجود(being)	و(and)	هيچگونه(any)	هيچكدام(none)
	يكي(one)	يكديگر(each other)	يك(a,an)	يا(or)	وي(he,she)

In an IR system all search strategies are based on comparison between the query and the stored documents. We implemented Boolean search strategy in our IR system. Next we asked computer students to make a query, and then determine which documents in the collection are relevant to their query. Table 2 shows the evaluation of bypassing or using Bon in our IR system.

Table 2- comparison of retrieval effectiveness with and without using Bon stemmer

	Recall	**Precision**
Without stemming	0.3595258	0.8974702
Using Bon stemmer	0.5421372	0.8397220

We see that stemming on a collection of Persian texts in the domain of computer science could increase recall by 40 percent. Bon takes each term in the query, and tries to determine which other terms in the database might have the same stem. If any possibly related terms are found, Bon presents them to the searcher for selection. It also allows searchers to focus their attention on other search problems. To illustrate how a stemmer is used in searching, consider the following example:

Example1: suppose a searcher has requested the following query:

(šabake ***yä*** *internet)* ***va*** *amnyat*   *(شبکه* ***يا*** *اينترنت)* ***و*** *امنيت*

(Network ***OR*** Internet ***) AND*** security

If the IR system doesn’t have stemming, the following text, for example, wouldn’t be retrieved. However this text is relevant to the query.

يكي از شايعترين پروتكلهاي توزيع كليد در حال حاضر پروتكلهاي KeyptoKnight است كه در اوايـل دهـه جـاري ميلادي ارائه گرديده است. در اين مقاله پس از معرفي خانواده پروتكلهاي مذكور به ارزيابي آن اقدام مي‌شود. هـدف از اين ارزيابي پاسخ مشروحي به اين سؤال است كه آيا KeyptoKnight به اهداف اعلام شده طراحـان خـود دسـت يافته است. در همين راستا يك مقايسه تحليلي بين اين پروتكل با پروتكلهاي مطرح و همسنگ آن انجام مي‌گيـرد. بالاخره ضمن معرفي يك رخنه در اين خانواده نشان داده مي‌شود كه اهداف طراحان در تـلاش بـراي دسـتيابي بـه حداكثر سازگاري با انواع توپولوژيهاي ايجاد *شبكه* به ايجاد خلل در جنبه‌هاي *امنيتي* پروتكل منجر شده است.

*** particle as a

**** part of “ر

***** particles as

> But if stemming is used during indexing time, the word “امنيتي”(amnyaty) is indexed as “امنيت”(amnyat). So this text would be retrieved in response to the above query.

Also Running Bon revealed that the Bon stemming algorithm is fast. Stemming a word on a PentiumIII 550 machine with 64Mbytes memory, takes only 0.03 seconds on average.

5. Conclusions

This paper describes first attempt on stemming Persian words. In Persian there are many exceptions for making any stemming rule. We have considered these exceptions in designing our stemmer. This stemmer that is called Bon, is a program that relate morphologically similar indexing and search terms in Persian texts.

Experiments revealed that using stems as index terms gives better retrieval results than using full words. In experiments of this paper Bon have increased recall parameter by 40 percent.

For testing Bon, we have gathered a Persian corpus (PCA). It seems appropriate to use the collection gathered in this paper to test other future Persian IR systems. In the future we will study other Persian IR models.

References

[1] Frakes W. B., *Stemming Algorithms*, http:// matrix.nbu.bg/books/books/book5/chap08.htm

[2] Anvari H., and Ahmadi Geavi H. *Persian Grammer,* Fatemi , Tehran, 1995.

[3] Salton G. and Mc Gill M. J., *Introduction to Modern Information Retrieval*, Mc Graw Hill, New York, 1983.