Bon: The Persian Stemmer

Source **Lecture Notes In Computer Science; Vol. 2510** archive
Proceedings of the First EurAsian Conference on Information and Communication Technology table of contents
Pages: 487 - 494
Year of Publication: 2002
ISBN:3-540-00028-3

Authors Masoud Tashakori
Mohammad Reza Meybodi
Farhad Oroumchian

Publisher Springer-Verlag London, UK

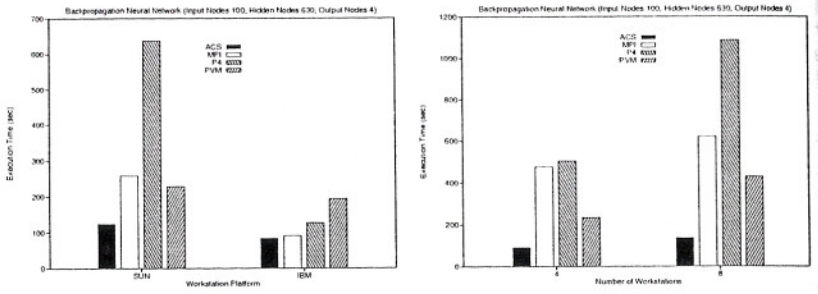

Fig. 4. Comparison of Application Performance

of the improvements of ACS are due to the overlapping of communication and computation and the tree-based broadcasting primitive.

5 Conclusion

In this paper, we have outlined the software architecture of a multithreaded message-passing system, ACS, and presented how ACS architecture can be applied to provide flexible and application-level group communication services. We have evaluated the performance of ACS group communication services and showed that ACS outperforms other message-passing systems. It is clear that the ACS novel architecture, which separates the data and control transfer and tree-based multicasting scheme played an important role in improving the performance of the communication primitives and the ACS applications.

References

1. S. Park and S. Hariri, "ACS: An Adaptive Communication System for Heterogeneous Wide-Area ATM Clusters", *Cluster Computing Journal*, pp. 229–246, 1999.
2. R. Butler and E. Lusk, "Monitors, message, and clusters: The p4 parallel programming system", *Parallel Computing*, Vol. 20, pp. 547–564, April 1994.
3. V. S. Sunderam, "PVM: A Framework for Parallel Distributed Computing", *Concurrency: Practice and Experience*, Vol. 2, No. 4, pp. 315–340, December 1990.
4. MPI Forum, "MPI: A Message Passing Interface", *Proc. of Supercomputing '93*, pp. 878–883, November 1993.
5. R. Renesse, T. Hickey, and K. Birman, "Design and performance of Horus: A lightweight group communications system", Technical Report TR94-1442, Cornell University, 1994.
6. L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia and C. A. Lingley-Papadopoulos, "Totem: A Fault-Tolerant Multicast Group Communication System", *Communications of the ACM*, Vol. 39, No. 4, pp. 54-63, 1996.
7. D. Dolev and D. Malki, "The Transis Approach to High Availability Cluster Communication", *Communications of the ACM*, Vol. 39, No. 4, pp. 64–70, 1996.

Bon: The Persian Stemmer

Masoud Tashakori[1], Mohammadreza Meybodi [2], and Farhad Oroumchian[3]

[1] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
Tashakori@noarvar.com

[2] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
Meybodi@ce.aku.ac.ir

[3] Electrical & Computer Engineering Department, Tehran University, Tehran, Iran
Foroumchian@acm.org

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 a 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.

1 Introduction

Morphological analysis is part of natural language processing and linguistic. Suffix stripping and stemming is part of computational morphological analysis. 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 [3].

In the field of information retrieval many experiments have been conducted to determine the value of stemming in retrieval process. There is a verity of methods to build a stemmer. The widely used Porter's algorithm [1] is a rule-based system, which iteratively removes suffixes. Porter algorithm does not guarantee correct form of the words to be produced after stemming. However, his algorithm is consistent and it is shown that it increases recall by up to 15%. Karaaj et al [2] have demonstrated that Porter's algorithm compresses the index vocabulary by about 43% on English text.

David Hull [4], Harmann [5] and others almost all agree that in information retrieval (IR), stemmers play an important role. In IR using stemmed words instead of the original words, could reduce the size of vocabulary. Since a single stem typically corresponds to several full terms, by storing stems instead of terms, compression factors of around 40-50 percent can be achieved. Thus in this paper we report on the development of a Persian stemmer which is called Bon. In next section we will look

M.H. Shafazand and A M. Tjoa (Eds.): EurAsia-ICT 2002, LNCS 2510, pp. 487–494, 2002.

at some of the properties of Persian words. Section 3 presents the Bon algorithm and section 4 describes the experiment. The last section is the conclusion.

2 Persian Words

Persian is an Indo-European language. As so in this language also new words can be constructed by adding prefixes and suffixes to base forms of words. Bon is an affix removal stemmer. Affix removal algorithms remove suffixes and/or prefixes from terms leaving a normalized form of the word. These algorithms sometimes also transform the resultant word into the real linguistic stem. A simple example of an affix removal stemmer is one that removes the plurals from terms [3].

Persian verbs have inflectional property, because they include person, number, and tense. For example, the verb "مي روم" (*mi-ra-vam*) which means, "I am going", consists of three parts: "م + رو + مي" (*mi + ro + m*) that is "I + go + ing" all in one word. Moreover, the infinitive verbs in Persian can be simple, or compound, or phrasal. There is at least one space between components of a compound or phrasal infinitive. For example, the verbs "قسم خوردن"*(ghasam xordan)* that means " to oath", and " از دست دادن" *(az dast dädan)* that means "to lose" have two and three components each in that order. In order to stem these verbs all their components should be located and evaluated as one word. This problem is considered in the Bon's stemming algorithm.

In Persian plural nouns are made by adding "ان"*(än)* or "ها"*(hä)* to the end of nouns. But if any noun ends in a "ه"*(eh)*; then before adding "ان"*(än)*, "ه"*(eh)* transforms into "گ"*(ge)* as depicted in the figure 1(a). There exceptions also, such as nouns that end in with "ان"*(än)* but are not plural e.g. "قهرمان"*(ghahramän)*. 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(b). 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(c). Moreover some nouns that are adopted from Arabic language have irregular plural forms ("Mokassar") as shown in the figure 1(d) [6].

In Persian a pronoun can be attached to the end of a noun. But if the noun ends in an "ا"*(ä)*, or "و"*(u)*; then a "ي"*(y)* is inserted before attaching the pronoun. Examples of this case are the word ("پا"*(pä)* meaning: foot) → ("پايم"*(päyam)* meaning: my foot), or the word ("چاقو"*(chäghu)* meaning: knife) → ("چاقويش"*(chäghuyaš)* meaning: his knife). Also if a singular pronoun is added to the end of a noun and the noun ends in a "ه"*(eh)*; then an "ا"*(a)* is added to the end of the noun before the pronoun. Example of this case is ("خانه"*(xäneh)* meaning: house) → ("خانه‌ام"*(xäneham)* meaning: my house).

Bon utilizes a dictionary of infinitives and present tense of infinitives for exceptional cases. For stemming words that are adopted from Arabic language, either a rule based or table look up approach can be used. Bon uses the first method.

a	خواننده ← خوانندگان (*xänandeh → xänandegän*)		
b	روحاني ← روحانيون *ruhäny → ruhänyun*	مؤمن ← مؤمنين *momen → momenin*	تدارك ← تداركات *tadärok → tadärokä*
c	دوا ← دواجات (*davä → davägat*)	شيريني←شيرينيجات (*širini → širinijat*)	شور ← شورجات (*šur → šurjat*)
d	كتاب ← كتب (*ketäb → kotob*)		

Fig. 1. Different Cases of Plurals in Farsi (Persian)

3 Bon Algorithm

Bon uses an iterative longest-match stemming algorithm. 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. After all characters have been removed, the resulting stem may not be correct. For example the word "خانگي"*(xänegy)* that means, "home-made", may be reduced to the stem "خانگ"*(xäneg)* which is incorrect form of the real stem "خانه"*(xäneh)* "house". There are two techniques to handle this: re-coding or partial matching [3].

Re-coding 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, one might say that two stems are equivalent if they agree in all but their last character [3]. Bon benefits from re-coding technique.

Bon has four major components:

1. Stemming rules that are extracted from Persian word construction rules.
2. A dictionary of Persian infinitives.
3. A dictionary of "Mokassar" words and their singular form
4. A dictionary of Persian roots.

Because of difficulties in building the last dictionary, an experimental version with almost 7000 words was built from the collection. In order to construct this dictionary, words were extracted from 450 abstracts in our collection. Then by running Bon, words that were derivative of any other word in the dictionary were eliminated. Moreover we gradually added or eliminated many words to/from roots dictionary.

In writing Persian, some of the letters of the words are attached together and some are separated. For example in word "خوردن"*(xordan)* that means eating, the first two letters are attached together but the last three letters are separated. A word boundary detection program finds the boundary of the words by looking simultaneously at each letter and its followings letters.

As depicted in the figures 2 and 3 the Bon algorithm has two major procedure: *Stem()* and *AffixRemove()*. The *Stem()* function takes a word and returns the stem of

the word. This function first, checks whether the word could be a verb or not. If the word is a verb then *Stem()*will return the infinitive of the verb. But if the word is not a verb, *Stem()* will search the word in Mokassar dictionary. If the word is found in the dictionary, its corresponding singular form is returned as the stem of the word. Now, if the word is not a verb or a Mokassar plural noun, then the word is turned over to the *AffixRemove()* function.

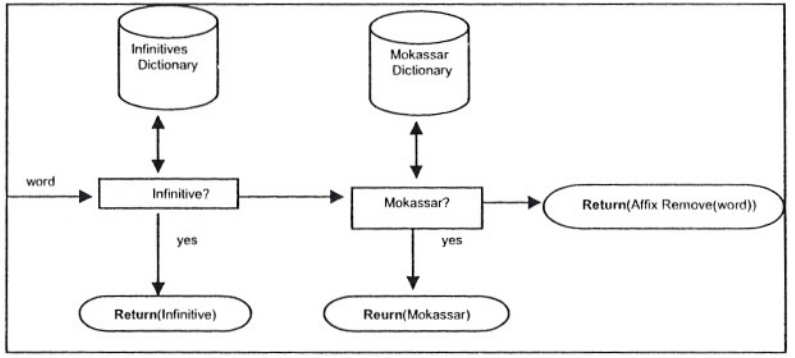

Fig. 2. Bon Algorithm: Stem() procedure

The *AffixRemove()* function starts with removing first (longest) possible affix from the word. If a possible affix could not be find, *AffixRemove()* will return the input word. But in the case of removing a possible affix, the stripped word is searched in the Persian roots dictionary. If the stripped word exists in this dictionary, *AffixRemove()* will return the stripped word. Otherwise it examines the possibility that the stripped word is a verb or Mokassar plural noun. Finally if this possibility fails, the word is restored and the *AffixRemove() function* will try to remove another affix until no more affixes could be found.

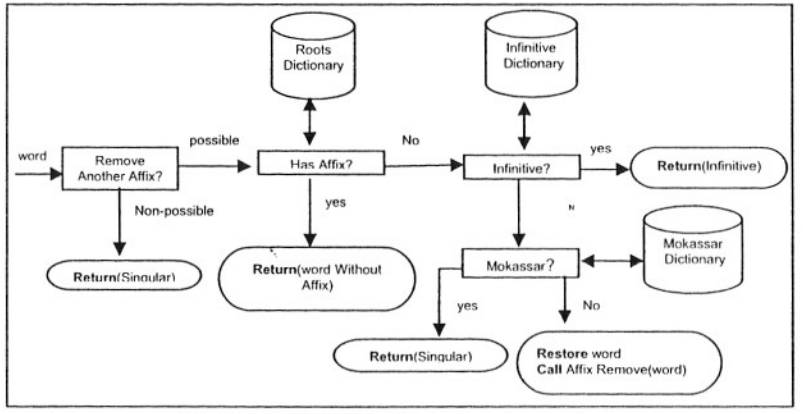

Fig. 3. Bon Algorithm: Affix remove() procedure

The reason for using Persian roots dictionary is that in some cases although the word contains the sign of an affix but it is actually a root word and should not be stemmed. For example the word "بیرون" (*birun*) ends with an "ون" (*un*) which is a sign of plural. But this word is a root word and not a plural. By having the root forms in the Persian roots dictionary, mistakes such as this would be prevented.

Next section describes experiments of running Bon on a Persian text collection.

4 Experiments

There are many ways to evaluate the performance of a stemmer. Paice has used two parameters in evaluating different stemming algorithms in English [7]. The Paice's method is independent of the context where the stemmer is going to be used. David Hull uses the context of IR to evaluate different stemming algorithms [4]. Kraaij et al have a good review of the evaluation methods of stemmers [2]. For this experiment, we looked at the improvements that can be achieved by using stems instead of the original words in a Persian information retrieval system. This is a popular method of evaluation used by Porter, David Hull and others. The retrieval effectiveness can be measured by recall and precision [8].

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.

Persian word boundary detection was applied first and the Persian words have detected. Then a stop list of 150 high frequency Persian words was used to filter out words with very low discrimination value. Table 1 lists our Persian stop list and their meanings.

For this experiment a Boolean retrieval system was implemented. Then computer students made up 32 queries, and then went through the collection and determined which documents are relevant to their query. To evaluate the Bon's performance, first the queries were presented to the system and the output were taken and recall and precision of each query was calculated based on the students relevance judgments. Next, Bon was used to present the students with all the terms that shared the same stem as the query words. Students could then expand their query by these newly introduced words. Then the expanded queries were run through the system again and the precision and recall of their output was calculated again. The precision and recall values were averaged over all the queries. Table 2 shows the difference in precision and recall with and without query expansion by using Bon stemming algorithm.

As depicted in table 2, query expansion by using Bon can increase the recall by 40 percent. There is also a slight drop in precision but this drop is more than offset by the increase in recall. To illustrate how a Persian stemmer such as Bon can help in searching, consider the following example:

Table 1. Typical stop list for a Persian IR System

ان(that)	آنان(they)	آنجا(there)	آنچه(whatever)
آنكه(which)	آنگاه(then)	آنها(those)	از(from)
است(is)	اگر(if)	اگرچه(although)	الان(now)
اما(but)	انجام(do)	او(he,she)	اي*
ايشان(they)	اين(this)	اينجا(here)	اينست(it's)
اينگونه(so)	اينها(these)	با(with)	باشد(be)
بايد(must,should)	بدون(without)	بر(upon,over)	براي(for)
بنابر**	بنابراين(therefore)	به(at)	بيشتر(more)
بين(between)	پس(then)	تا(until)	تايي(many)
تو(you)	توسط(via)	چرا(why)	چگونگي(manner)
چگونه(how)	چنان(that)	چنانچه(if)	چند(a few)
چندان(so much)	چندگانه(multiple)	چندين(many)	چنين(this)
چه(what)	چون(since)	چيز(thing)	چيزي(anything)
چيست(what is)	حتي(even)	خواهد(will)	خواهيم(will)
خود(itself)	خودت(yourself)	خودتان(yourselves)	خودش(himself,herself)
خودشان(themselves)	خودم(myself)	خودمان(ourselves)	خويش(self)
داده(given)	داراي(having)	دارد(has)	داشته(having)
در(in)	درباره(about)	دو(both)	ديگر(other)
ديگران(others)	ديگري(another)	را***	روي(on)
زيرا(because)	سپس(next)	شامل(including)	شايد(may)
شد(became)	شده(becoming)	شما(you)	شود(be)
صورت****	فقط(only)	كدام(which)	كرد(do)
كردن(do)	كه(that)	گرچه(though)	گرفته(taken)
لكن(however)	ليكن(however)	ما(we)	مابين(between)
مانند(as)	مختلف(several)	من(I)	مورد(case)
ميباشد(is)	ميتوان(can)	ميتواند(could)	ميدهد(give)
ميشود(becomes)	ميشوند(become)	ميكند(do)	ميكنند(do)
نظر(seem)	نميتوان(cannot)	نوع(kind)	نياز(need)
نيز(again)	نيست(isn't)	نيستند(aren't)	ها*****
هاي*****	هايي*****	هر(each)	هرچه(whatever)
هريك(every)	هست(is)	هستند(are)	هستيم(are)
هم(too)	همان(same)	همانطور(same)	همانند(like)
همچنين(also)	همچون(like)	همديگر(each other)	همه(all)
همواره(ever)	هميشگي(usual)	هميشه(always)	همين(this)
هنوز(still,yet)	هيچ(never)	هيچكدام(none)	هيچگونه(any)
و(and)	وجود(being)	وگرنه(otherwise)	ولي(but)
وي(he,she)	يا(or)	يك(a,an)	يكديگر(each other)
يكي(one)			

* This word is part of Persian present perfect verbs for the second person.
** part of "بنابر اين" (therefore)
*** particle as a sign of the definitive direct object
**** part of "بدين صورت" (so)
***** particles as plural sign

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

Example1. Suppose a searcher has requested the following query:

Persian Query: شبكه يا اينترنت) و امنيت (*šabake yä internet) va amnyat*)

Meaning in English: (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 اوايل دهه جاري ميلادي ارائه گرديده است. در اين مقاله پس از معرفي خانواده پروتكلهاي مذكور به ارزيابي آن اقدام مي‌شود. هدف از اين ارزيابي پاسخ مشروحي به اين سوال است كه آيا KeyptoKnight به اهداف اعلام شده طراحان خود دست يافته است. در همين راستا يك مقايسه تحليلي بين اين پروتكل با پروتكلهاي مطرح و همسنگ آن انجام مي‌گيرد. بالاخره ضمن معرفي يك رخنه در اين خانواده نشان داده مي‌شود كه اهداف طراحان در تلاش براي دستيابي به حداكثر سازگاري با انواع توپولوژيهاي ايجاد **شبكه** به ايجاد خلل در جنبه‌هاي ***امنيتي*** پروتكل منجر شده است.

But as it is highlighted in the text the word "امنيتي" (*amnyati*) which is from the same stem as the word "امنيت"(*amnyat*) is present in the text. Therefore using a stemmer such as Bon can increase the recall by retrieving documents that could have been missed otherwise.

Also the Bon stemming algorithm is fast. The speed of stemming measured for all the words in the collection and averaged. Stemming of a word on a Pentium III 550 machine with 64Mbytes memory takes only 0.03 seconds on average.

5 Conclusions

This paper describes a Persian stemmer called Bon. Persian stemming rules have many exceptions that are considered and handled in designing this stemmer. Experiments revealed that using stems as index terms gives better retrieval results than using full words. In the experiment reported here, Bon have increased recall by 40 percent.

A by-product of our experiment is building a small Persian corpus (PCA). It seems appropriate to use this collection in future experiments on Persian retrieval models.

References

1. Porter M. F., "An Algorithm for Suffix Stripping", Program, vol. 14, no. 3, pp. 130-137, 1980.
2. Kraaij W., Pohlmann R., "Evaluation of A Dutch Stemming Algorithm", The New Review of Document and Text Management, vol. 1, pp. 25-43, 1995.
3. Frakes W. B., Stemming Algorithms, available at: http:// matrix.nbu.bg/books/books/book5 /chap08.htm
4. Hull David A., "Stemming Algorithms: A Case Study for Detailed Evaluation", Journal of The American Society For Information Science, vol. 47, no. 10, pp. 70-84, 1996.
5. Harman D.,"How effective is suffixing?", Journal of the American Society for Information Science, vol. 42, no. 1, pp. 7-15, 1991.
6. Anvari H., and Ahmadi Geavi H. Persian Grammer, Fatemi, Tehran, 1995.
7. Paice C. D., "An Evaluation Method For Stemming Algorithms", Proceedings of ACM-SIGIR94, pp. 42-50, 1994.
8. Salton G. and Mc Gill M. J., Introduction to Modern Information Retrieval, Mc Graw Hill, New York, 1983.

Current and Future Features of Digital Journals

Harald Krottmaier

Institute for Information Processing and Computer Supported new Media (IICM)
Inffeldgasse 16c, A-8010 Graz, Austria
hkrott@iicm.edu

Abstract. Features in currently available systems are often restricted to passive 'consumption' of articles stored in the corresponding digital journal. This paper classifies features into two categories (overall system structure, and content based features) and gives an outlook of planned implementations in the Journal of Universal Computer Science (J.UCS). It also shows that using the powerful Hyperwave Information Server (HIS) makes it quite easy to *implement* features and make knowledge management features (such as 'find an expert on a topic') available to users of a digital journal.

1 Introduction and Overview

First generation publishing systems were very static and inflexible. Simple pages encoded in some digital format were prepared by some editorial team, stored on a server system, and then transfered to the user on request. Systems were based on ordinary web-servers without any interactive features but 'get document XXX' were very often used as base system. With ongoing development of technologies user supporting tools like 'search through the content stored on a server' etc. were implemented. This paper is about features available in digital journals. The following systems were explored in detail:

ACM-DL: Digital library of the ACM (Association for Computing Machinery) [ACM Digital Library, 2002].
LINK: Information service published by Springer [LINK, 2002].
Xplore: Service published by IEEE (Institute of Electrical and Electronics Engineer) [IEEE-xplore, 2002].
ScienceDirect (SD): published by Elsevier [Science Direct, 2002]
JoDI: The Journal of Digital Information maintained by the IAM Research Group, University of Southampton [JODI, 2002].
JUCS: The Journal of Universal Computer Science. A publication of the Know-Center in cooperation with Springer Co.Pub., JOANNEUM RESEARCH and the IICM, Graz University of Technology [J.UCS, 2002].

To simplify the following discussion, we use the bold printed keys to refer to the systems, i.e. we write *Xplore* when we talk about the digital library system of the IEEE. *ACM-DL*, *LINK*, *Xplore* and *ScienceDirect* are serving

A Min Tjoa (Eds.)

EurAsia-ICT 2002: Information and Communication Technology

LNCS 2510

First EurAsian Conference
Shiraz, Iran, October 2002
Proceedings

Springer