

A Stemming Algorithm for the Farsi Language

Kazem Taghva, Russell Beckley, and Mohammad Sadeh
Information Science Research Institute
University of Nevada, Las Vegas
taghva@isri.unlv.edu

Abstract

In this paper, we report on the design and implementation of a stemmer for the Farsi language. The results of our evaluation on a small Farsi document collection shows a significant improvement in precision/recall over not stemming.

1 Introduction

Farsi is an Indo-European language, spoken and written primarily in Iran, Afghanistan, and a part of Tajikistan. It is associated with Persian culture and is often called *Persian*. To facilitate the information retrieval component of our Farsi search and display technology project[6], we designed and implemented a Farsi language stemmer.

To stem a word is to find a more general form of it, possibly its root. For example, stemming the term *interesting* may produce the term *interest* or *interes*. Though a stemmer might not always give the root, we want all words that have the same stem to have the same root. On the other hand, for information retrieval, we do not always want all words with a given root to have the same stem because some words with the same root may be topically uncorrelated e.g. *preside* and *president*. There is much research of the effects of stemming on searches of English document collections. Stemmers such as the Lovins and Porter stemmers sometimes improve precision/recall scores. [2] However, they only stem English terms.

Like English, Farsi has an affixitive morphology. In other words, suffixes and prefixes are concatenated to words to modify meaning. Farsi is read from right to left, so that prefixes are attached to the right of the root, and suffixes are attached to the left. Like English nouns, Farsi nouns are modified to signify possession, agency, and plurality. However, Farsi verbs are modified more extensively than English verbs. Farsi verb forms vary according to tense, person, negation, and mood. The dozens of variations of each Farsi verb are a primary motivation for our stemmer.

Infinitive	Imperative Mood	English Translation
آوردن	آور	to bring
پرسیدن	پرس	to ask

Table 1. Regular Infinitives and Imperative Moods

This paper consists of five sections in addition to this Introduction. Section two is a short overview of Farsi and its morphology. Section three describes our stemming algorithm, Section four describes our implementation, Section five evaluates the stemmer according to precision/recall scores, and Section six discusses our conclusion and future work.

2 Farsi Language

Farsi verbs usually derive from their imperative forms. Hence, from a linguistic point of view, the first step in extracting the root is to find the imperative mood of the word. For example, we can find the root of شنونده ("listener") by removing the suffix نده ("one who does").

Regular infinitives end with the suffix دن and the imperative mood of the regular infinitives is found by removing the last two or three characters. Examples of regular infinitives and their imperative moods are listed in Table 1.

However, it is difficult to find the imperative mood for *irregular* infinitives, as they have no regular pattern. Examples of irregular infinitives and their imperative moods are listed in Table 2.

The imperative form of irregular infinitives is based on how the words are heard or used and is known as *سماعی* (pronounced "sama'i").

In Farsi, it is common to add the prefix ب (sounds like English "b") and ن ("n") for positive and negative moods respectively. So, for example برو ("go") and نرو ("do not go") are the positive and negative forms of رو.

With the imperative form of a verb, one can generate

Infinitive	Imperative Mood	English Translation
رفتن	رو	to go
کردن	کن	to do

Table 2. Irregular Infinitives and Imperative Moods

Suffix	Plural	English Translation
پسر	پسرها	sons
جوان	جوانان	young people
مشکل	مشکلات	diculties

Table 5. Plural Suffixes

Suffix	Present Tense	English Translation
م	بروم	I go
ی	بروی	You go
ه	برود	He goes
یم	برویم	We go
اید	بروید	You go
ند	بروند	They go

Table 3. Present Tense Suffixes

tenses such as present tense. For example, to generate the indicative present tense (مضارع اخباری) for the verb رفتن ("to go"), we start with the positive imperative برو and add the appropriate suffixes as found in Table 3.

The present tense rules are generally used to generate other tenses. Past tense is generated from the infinitives by removing the character ن and adding the same suffixes as above. The past tense (زمان گذشته) of the verb رفتن is رفت and its variations are listed in Table 4.

It should be noted that no suffix is added for past tense singular third person; this is due to the fact that if we add the suffix د, then the pronunciation becomes awkward. Readers interested in various forms of Farsi verbs are referred to [1].

The plural forms of nouns are formed by adding the suffixes ها, ان, and, for words borrowed from Arabic, ات. Table 5 shows examples of plurals.

Farsi has a well defined and detailed morphology of which the above description gives only a flavor.

Suffix	Past Tense	English Translation
م	رفتم	I went
ی	رفتی	You went
	رفت	He went
یم	رفتیم	We went
اید	رفتید	You went
ند	رفتند	They went

Table 4. Past Tense Suffixes

3 The Algorithm

The Farsi stemmer is similar to the Porter stemmer[3]. Both are based on morphology. Also, both stemmers search for certain suffixes and use multiple phases conforming to the rules of suffix stacking. Furthermore, they both enforce a lower bound on the information a stem retains. However, there are important differences. For example, the Porter stemmer identifies patterns of consonants and vowels to estimate the information content; in Farsi, many spoken vowels are not written, so the Farsi stemmer uses stem length to define a lower bound on information content (currently, the minimum stem length is three). This limit is crucial when a non-suffix substring of a short word is incorrectly identified as a suffix. Another difference is that the Farsi stemmer identifies prefixes while the Porter Stemmer does not.

The first step of the stemmer algorithm is to find a terminal substring of the input word that is in a list of common Farsi morphological suffixes. If multiple suffixes match the word, the stemmer chooses the longest suffix that would leave a stem with three or more characters. Consider the Farsi word دستشان ("their hands"). Both the plural suffix ان and the plural possessive شان match the end of the word. Removing ان leaves four letters, and removing شان leaves three letters. Because both leave long enough stems, the stemmer removes شان, the longest, giving دست (hand).

The suffixes are grouped as *verb-suffixes*, *plural-noun-suffixes*, *possessive-noun-suffixes*, *other-noun-suffixes* (e.g. نده), and *other-suffixes* (e.g. تر). This grouping guides removal of prefixes from verbs and removal of multiple suffixes from a noun. If the stemmer first identifies the suffix ند in the word نرفتند ("they did not go") as a verb-suffix, it then identifies and removes the prefix ن to produce the stem رفت ("went").

Noun suffixes are stacked according to the pattern (reading right-to-left):

$$\{possessive\}\{plural\}\{other\} < stem >$$

For example, the stemmer first finds the possessive noun suffix یمان in the word خواننده هایمان ("our singers"), then it finds the plural noun suffix ها, and, finally, it finds the other-noun-suffix نده (which signifies agency) to give the stem خوان ("sing"). Hence the stemmer removes up to three suffixes from nouns.

state	ا	ت	س	م	ن	ه	ی	else
0	12	12	12	2	1	12	12	12
1	3	12	12	12	12	12	12	12
2	12	12	12	12	12	12	4	12
3	8	5	8	8	8	8	8	8
4	6	10	10	10	10	10	10	10
5	9	9	12	9	9	9	9	9
6	10	10	10	10	10	7	10	10
7	11	11	11	11	11	11	11	11

Table 6. Two-dimensional array representing the DFA in Figure 1

In addition, there are some unusual cases. Usually, when the stemmer finds the suffix *تان*, it removes it. However, when it is preceded by *س* it ignores the suffix, because the Farsi suffix *ستان* ("location of"; pronounced "stan") is often used for countries and regions, e.g. "Kurdistan.". The stemmer does not remove *ستان* because we believe, generally, the resulting conflation (e.g. *Kurd = Kurdistan*) are not helpful for a search engine.

Another exception is that the stemmer finds verbal suffixes *د* and *ت* but does not remove them. It was explained in Section 2 that the infinitives end with *دن* or *تن*. Most of the Farsi tenses are formed after removing the suffix *ن* but leaving characters *د* or *ت*.

In many cases, the stemmer looks at the letter preceding a supposed suffix. Often, this pre-suffix can be used to determine whether the match is actually a suffix and, if it is, whether it ought to be removed. In such cases, if the suffix is removed, the pre-suffix remains.

4 Implementation

To implement the algorithm, the Farsi stemmer uses a 70-state Deterministic Finite Automata (DFA). The DFAs input string is obtained by reversing the stemmers input string. Figure 1 depicts a portion of the machine which finds suffixes *تان*, *ان*, *هايم*, and *ين*. The design of the machine incorporates the minimum stem length requirement and all pre-suffix requirements.

The DFA is encoded as a two-dimensional array. The rows represent states and the columns represent input letters. Table 6 shows such a two-dimensional array representing the machine in figure 1.

The DFA driver starts from the end of the stemmers input word and works toward the third letter from the front. The DFA never sees the first two characters of the word. In each round the driver determines the next state by observing the entry at row *s* and column *l*, where *s* is the current state and *l* is the input character. When the machine reaches a final

state	SuffixGroup[state]
0	NIL
1	NIL
2	NIL
3	NIL
4	NIL
5	PL2
6	VB2
7	VB2
8	PL2
9	PO3
10	VB2
11	VB4
12	NIL

Table 7. SuffixGroup[] for the machine in Figure 1

state, the word and the state number is fed to a post processor for suffix removal. For example, if we want to stem the word *زمان* ("times"), we remove the first two characters *زم* and feed the remaining characters *ان* to the DFA. Readers can observe that the DFA will end in state 3, which is a NIL state; in every case, when the only possible suffix would leave too short a stem, the final state will be NIL. On the other hand, stemming the word *پسران* ("boys"), will return state 8.

The post processor uses the final state to determine a suffix group. `SuffixGroup[]` is a one-dimensional array. If *F* is the final state, `SuffixGroup[F]` gives the identifier for the suffix group. This identifier is used to strip the suffixes. For example, in the case of our first word *زمان*, `SuffixGroup[3]` returns NIL which means no suffix will be stripped from this word (the stem would be the correct root, but it would be too short). In the case of the word *پسران*, the `SuffixGroup[8]` returns PL2, which signifies the word is plural and two characters should be removed. Hence the stemmer will return *پسر* ("boy"). Table 7 shows a suffix group array for the DFA in Figure 1.

If the identified suffix is possessive or plural, the post processor will feed the stripped word to the DFA to identify other suffixes. If subsequent suffixes are found, the post processor removes them only if they belong to the correct noun class or classes. When the post processor identifies a verbal suffix state (such as 10 in the figure), it feeds the stripped word to a prefix DFA which is similar to our suffix DFA.

5 Evaluation

To evaluate the Farsi stemmer, we observed its effect on precision/recall using our Farsi information retrieval system (a vector-based system) [4], a fixed set of Farsi queries, and a fixed document collection.

A collection of 1647 Farsi documents, primarily internet documents, was created. Native Farsi speakers compiled a list of sixty queries. For each document in the collection, and for each query, it was determined whether the document was relevant to the query. The Farsi collection was indexed without using the stemmer, and without removing stopwords [5]. We then processed each query in the list. The identical procedure was repeated except that the stemmer and the stopword list were introduced.

11-point average precisions are given in following tables. The test in which the stemmer was used shows an increase of .033, or 18 better.

recall	precision	interpolated
0	0.328	0.483
10	0.253	0.353
20	0.228	0.273
30	0.119	0.215
40	0.131	0.197
50	0.151	0.170
60	0.078	0.105
70	0.055	0.074
80	0.039	0.060
90	0.027	0.049
100	0.046	0.046
Average	0.132	0.184

Table 8. Average precision/recall results using no stemmer and no stopword removal

recall	precision	interpolated
0	0.342	0.544
10	0.310	0.413
20	0.290	0.333
30	0.142	0.242
40	0.137	0.210
50	0.171	0.191
60	0.096	0.141
70	0.086	0.106
80	0.045	0.080
90	0.030	0.065
100	0.060	0.060
Average	0.155	0.217

Table 9. Average precision/recall results using stemmer and stopword removal

- [2] David A. Hull. Stemming algorithms a case study for detailed evaluation. Technical report, Rank Xerox Research Centre, Meylen, France, June 1995.
- [3] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130137, 1980.
- [4] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.
- [5] Kazem Taghva, Russell Beckley, and Mohammad Sadeh. A list of farsi stopwords. Technical Report 2003-01, Information Science Research Institute, University of Nevada, Las Vegas, July 2003.
- [6] Kazem Taghva, Ron Young, Jeffrey Coombs, Ray Pereda, Russell Beckley, and Mohammad Sadeh. Farsi searching and display technologies. In *Proc. of the 2003 Symp. on Document Image Understanding Technology*, pages 4146, Greenbelt, MD, April 2003.

6 Conclusion and Further Research

The results of our stemming test indicate that the Farsi stemmer improves retrieval. Our tests were done on a small collection, so the effect of the stemmer on bigger collections is not known at this time. There are many ways that the stemmer can be modified, including editing the list of suffixes, changing the minimum stem length, and foregoing prefix removal.

References

- [1] A. Gharib, M. Bahar, B. Fooroozanfar, J. Homaii, and R. Yasami. *Farsi Grammar*. Jahane Danesh, 2nd edition, 2001.

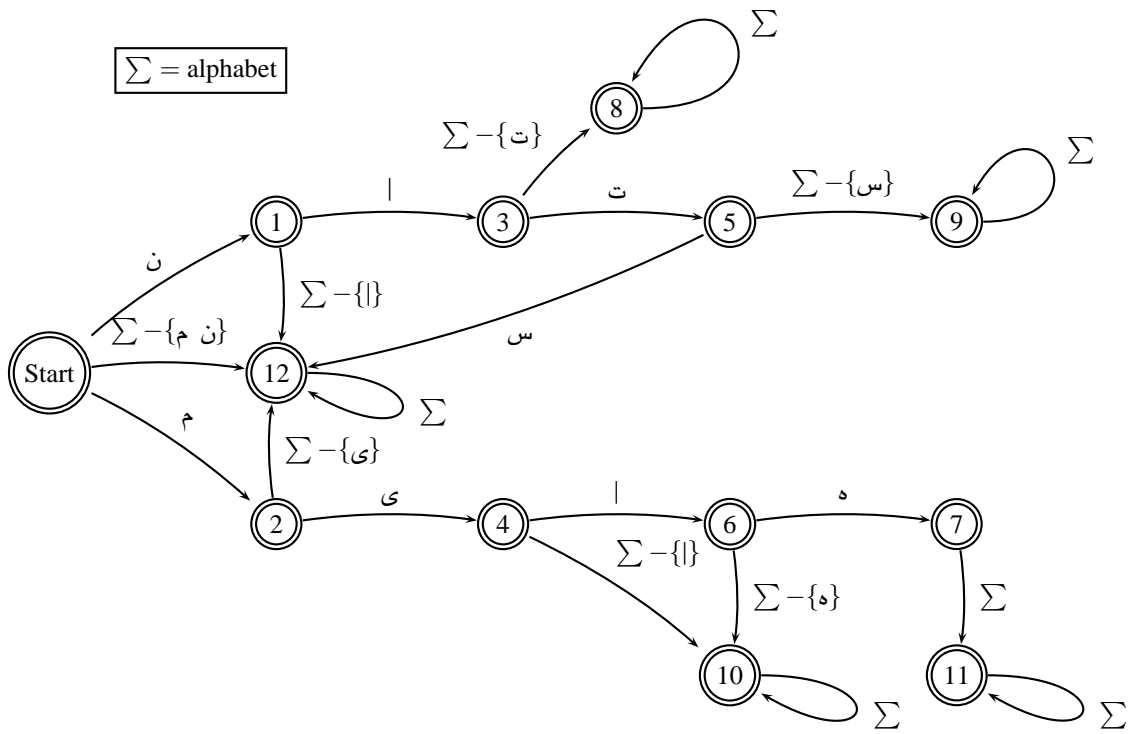


Figure 1. A part of the Farsi Stemming DFA