*International Journal of Information Science and Management*
*Vol. 13, No. 2, 2015, 77-91*

# Creating Appropriate Corpus for Information Retrieval and Natural Language Processing in Persian Language

**Zahra Abdolhosseini**
Department of Computer Engineering, Alzahra University, Tehran, Iran
zabdolhossini@gmail.com

**Mohammad Reza Keyvanpour**
Department of Computer Engineering, Alzahra University, Tehran, Iran
keyvanpour@alzahra.ac.ir

**Marjan Shabani Asl**
Department of Computer Engineering, Shariaty University, Tehran, Iran

## Abstract

Persian natural language processing (NLP) researchers have many limitations to access linguistic tools which are suitable for text processing. Therefore, research in Persian text processing is very limited. Since dataset is an important requirement for experiments and their evaluation, we aimed to create appropriate corpora for information retrieval and natural language processing in Persian. The provided corpora in this article are based on HAMSHAHRI dataset which is appropriate for simple information retrieval and simple natural language processing because it has not been tagged. We converted this dataset to tagged collection and increased its text quality. The new corpora minimize the text preprocessing requirement. Here we have used STep-1 tools for text processing and have proposed some ideas to remove the bugs of these tools in order to increase their quality. At the end we used the new corpora for text retrieval and results showed performance improvement.

**Keywords:** Persian Text Corpus, Persian Tagged Texts, Improved Hamshahri, Improved STeP1.

## Introduction

Persian natural language processing (NLP) researchers have many limitations for accessing linguistic tools which are suitable for text processing. Limitations include lack of free tagged dataset for Persian text processing and lack of full words dictionary. These limitations create many challenges for researchers. Actually one of the main reasons for low numbers of studies on Persian texts is low numbers of appropriate standard dataset to experiment proposed methods. But in many languages such as English many tools have been constructed for doing different works such as morphology analysis, POS Tagging, Spell

Checking, Chunking and the other (Shamsfard et al., 2010). Also many works have been done for corpora creation in other languages (Yang et al., 2004; Talvensaari et al., 2008; Miangah 2009).

Text processing of languages with limited resources such as Persian is complex and time consuming (Shamsfard et al., 2010). Therefore, researchers who work in this scope, create their tools in a limited domain. This problem limits the suitable tools for all applications in text processing scope. Therefore moving in the direction of creating these tools is very important. Since dataset is an important requirement for experiments and their evaluation, in this article we have tried to create appropriate corpora for information retrieval and natural language processing in Persian. Among few current Persian texts corpora, HAMSHAHRI collection (Aleahmad et al., 2009) is a standard dataset for experiment and evaluation methods. The provided corpora in this article are based on HAMSHAHRI collection which is appropriate for simple information retrieval and simple natural language processing because it has not been tagged. We converted this dataset to tagged collection and increased its text quality. The new corpora minimize the text preprocessing requirement. DOTIR (Darrudi et al., 2008) and BIJAN KHAN Corpora (Oroumchian et al., 2006) are the other Persian texts datasets which DOTIR is appropriate for information retrieval but has not been tagged and BIJAN KHAN is appropriate for only natural language processing but this dataset is not free. Notice produced corpora in this article are useful for both of Persian text retrieval and natural language processing. Here we have used STep-1 tools for text processing and have proposed some ideas to remove the bugs of these tools in order to increase their quality.

The rest of the paper is organized as follows. In Section 2 we presented more details of HAMSHAHRI corpora and STeP1 tools. In section 3 we expressed our system architecture and proposed methods for bugs removal from STeP1 and introduced structure of new corpus. Next we showed experiments results. Finally in section 5 conclusion is presented.

## Background

In this section we describe the details of used data and tools.

### HAMSHAHRI and the Other Persian Corpora

Up to now the works have been done for creating corpora in Persian information retrieval scope but HAMSHAHRI corpora is a standard collection which has more performance. Also Taghiyareh et al. (2003) used the text with size of 20MB but when dataset size is low, results are not reliable. Shiraz corpora (Amtrup et al., 2000) include the bilingual and tagged text with a size of 10 MB that has been constructed from Persian corpora to machine translation project in New Mexico State University. Also the other Persian dataset namely MAHAK has been produced for evaluation of Persian information retrieval system by Sheykh Esmaili et al., (2007). This dataset includes 3007 documents. So that MAHAK is not suitable for large information retrieval systems. Finally we describe the details of HAMSHAHRI corpora. HAMSHAHRI is one of the first online Persian newspapers in Iran; it has presented its archive to the public through its website since 1996. Creation of HAMSHAHRI corpora

started by Oroumchian et al. (2004) so that researchers employed a crawler to download available online news from the web site of HAMSHAHRI newspaper and presented some advantageous statistics of HAMSHAHRI corpora based on characteristics of the Persian language. This dataset were completed gradually until converted to current format. Previous version of the collection contains 58 queries that were not prepared based on TREC specifications but last version was prepared based on TREC specifications and these corpora were converted to standard corpus. The current version includes more than 160000 documents, 100 queries and their related judgments. The preparation steps of dataset were previously described (Oroumchian et al., 2004). Each document is shown by an ID, publication date, category, and text property. Figure 1 shows the sample of documents in this corpus.

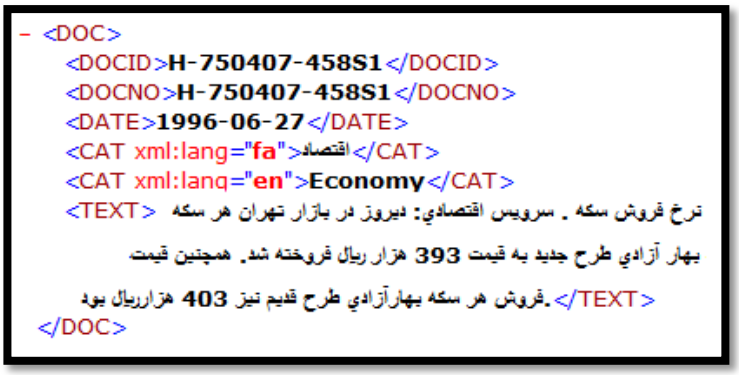


*Figure 1*: The sample of a document in HAMSHAHRI collection

**STeP-1 Tools**

Among the introduced tools for Persian text processing, STeP-1 is the first step in processing texts written in Persian language. Shamsfard et al. (2010) presented these tools. STeP-1 performs tokenization, morphological analysis and POS tagging. Users can select arbitrary combination of these services in different depths for their own task and application. In general it proposes the following activities for conversion of texts into a standard one.

1. Defining a computational standard script:
   - Adding short-spaces between different parts of a word (or a compound word).
   - b) Adding Spaces between words and phrases
   - c) Introducing the spacing rules between punctuations, numbers and special cases (ex. date)
   - d) Creating a lexicon with different spellings of words.
2. Converting texts to the standard script
   - Looking up in a dictionary
   - b) Checking the spelling
   - c) Correcting the spacing
     - replace white spaces with short spaces
     - Add white spaces (unknown words)

STep1 can be used as style correction and preprocessing tools in many natural language processing applications in Persian language. Also STep1 can be used as a stemmer. Text stemming is one of the works which are done for texts preprocessing. Stemming is a widely used method of word standardization designed to allow the matching of morphologically related terms (Tashakori et al., 2002). 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*.

Generally in natural language processing and other fields such as information retrieval (IR), stemmers play an important role. In information retrieval using stemmed words instead of the original words, could increase the level of the exhaustively of indexing, and could increase overall performance. Also stemming reduces the size of indexing files. Since a single stem typically corresponds to several full terms (Tashakori et al., 2002). (Karimpour et al., 2009) improved Persian information retrieval systems using stemming and part of speech tagging. Up to now some works are done for Persian texts stemmer. Such as (Berenjian, 2013) presented the Persian stemmer systems for stemming verbs and (Rashidi & Zolfy Lighvan, 2014) implemented a novel hierarchical Persian stemming approach. In the other work (Mehrad & Berenjian, 2011) provided a Persian language singular stemmer system. Also (Berenjkoob et al., 2009) presented a stemming method for Persian text summarization. (Estahbanati & Javidan, 2011) implemented a new stemmer for Farsi language with combination methods.

## Proposed Method

This section includes three subsections. In first subsection we present the classification of STep1 bugs and suggest the ideas for them and in second subsection describe our system architecture. In third subsection we introduce the structure of produced data.

### The Classification of STep1 Bugs and Removing Them

In this Subsection we describe STeP1 Bugs and introduce the ideas for removing them. Figure 2 shows the classification of these bugs.

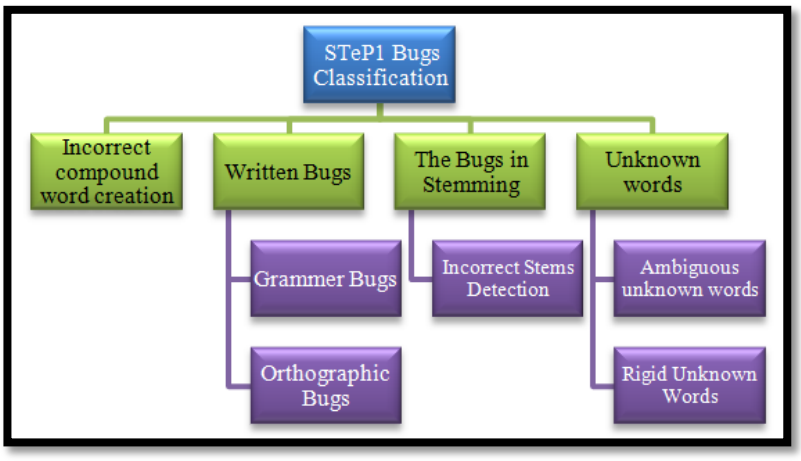


*Figure 2.* The classification of STep1 Bugs

**Incorrect Compound Word Creation**

We can correct the spaces between compound words by STeP-1. Also we can set the option that does not allow correction. If correction was done by STep1 tokenizer, a bug might occur. That bug is incorrect compound word creation. It occures when two successive and meaningful words are presented in text whose compound is meaningful. In these times STeP1 deletes the space between words and considers them as compound word whereas it is likely that there is no need to delete the space. For example in sentence of " ما هر سال به مشهد می رویم" (We go to Mashhad every year) STeP1 connects "ما" and "هر" and converts them to "ماهر". The produced word is an adjective in Persian language. We proposed a method For removing this bug. We can compare the words of the main text with the words of tokenized text which is the produced text after applying tokenizer on text, If there is not accordance between two words and the new word has been produced by merging current word and next word, it is possible that this bug has been occured. Next we compare frequency of produced word with frequency of two words in main text, if frequency of produced word is very lower than frequency of two words, we do not allow to merge words. However this idea is not strong and it is possible we don't detect this bugs correctly. The other idea is using the grammatical structure of the sentence. In the previous example there is no accordance between the new produced word and sentence verb. Then we can consider the new produced word as incorrect.

**Written Bugs**

There are two bugs in this category: grammatical bugs and orthographic bugs. For example if there is no accordance between subject and verb in the sentence, STep1 cannot correct it. Using tagged dataset is necessary for correcting grammatical bugs. The new produced corpus in this article helps perform the methods for removing this bug. Also STeP1 cannot correct orthographic bugs. There are other studies on orthographic bugs (Sheykholeslam et al., 2012; Rezvan et al., 2009).

**The Bugs in Stemming**

This category includes detection of incorrect stems bugs. There is a bug in STep-1 stemmer. STep-1 stemmer sometimes considers word stem by removing the suffix and prefix of word. This method creates a problem because in the Persian language the specified suffix and prefix may not be actually the suffix or prefix and they may be parts of the main word. For example in the word "ماشین"(car), "ین" is considered as suffix and "ماش" as stem. Whereas "ین" is part of main word and there is no suffix. We will describe the method for removing this bug in the next section.

**Unknown Words**

This category includes the bugs of ambiguous unknown words and rigid unknown words. When the STep1 tokenization tool is apply on texts, the spaces between words are corrected and text words are produced but this tool has some bugs. For example it corrects compound

words like "روز نامه" (newspaper) and writes its parts with no space or with half space but it cannot correct compound word if its parts be connected like "تخریبکنندگان" (the ruinous) and considers it as an unknown word. Also this tool sometimes cannot detect a word because there isn't any space between two independent words, in this time it produces different forms of word. For example "VaGhodrat,وقدرت " can be split to [وقدرت، و قدرت، و قد رت]. This problem is the other bug. Therefore we categorized these bugs to rigid unknown words and ambiguous unknown words. In other subsection we will describe the ideas for removing these bugs.

## Proposed System Architecture

This subsection describes our system architecture. Our system includes five steps. In this system input data format is xml and first we had to extract text elements from xml files. After text extraction we applied STeP-1 tokenization on text. Next STep1 tokenization and stemmer is improved. After improving tools and applying them on texts, we saved the results as xml file. The encoding of the produced files is utf-16; therefore, it is necessary that we convert it to utf-8. Also we again corrected the spaces between punctuations and words at the end. Further we will describe the third and fourth steps. Fig 3 shows these steps.

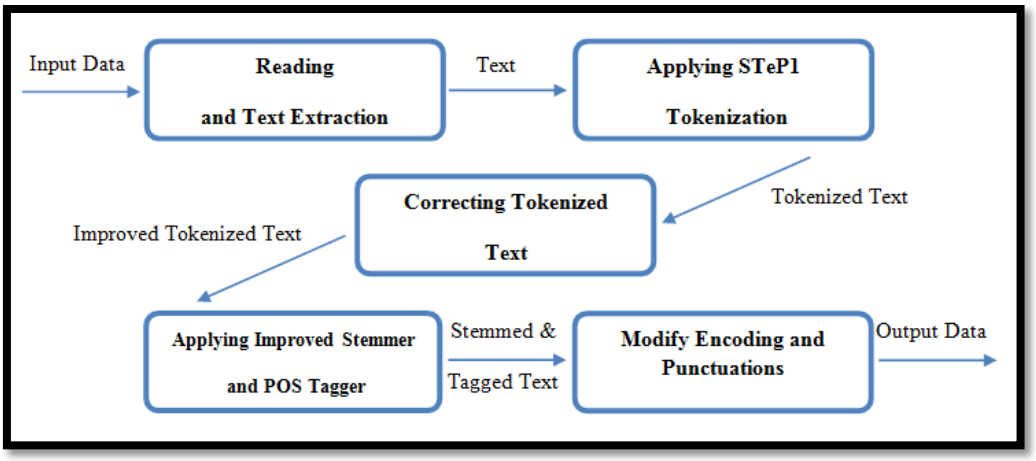


*Figure 3.* The proposed System Architecture

## Correcting Tokenized Text

Because of above bugs we use the following methods for removing these bugs. Figure 4 shows the flowchart of correcting tokenized text.

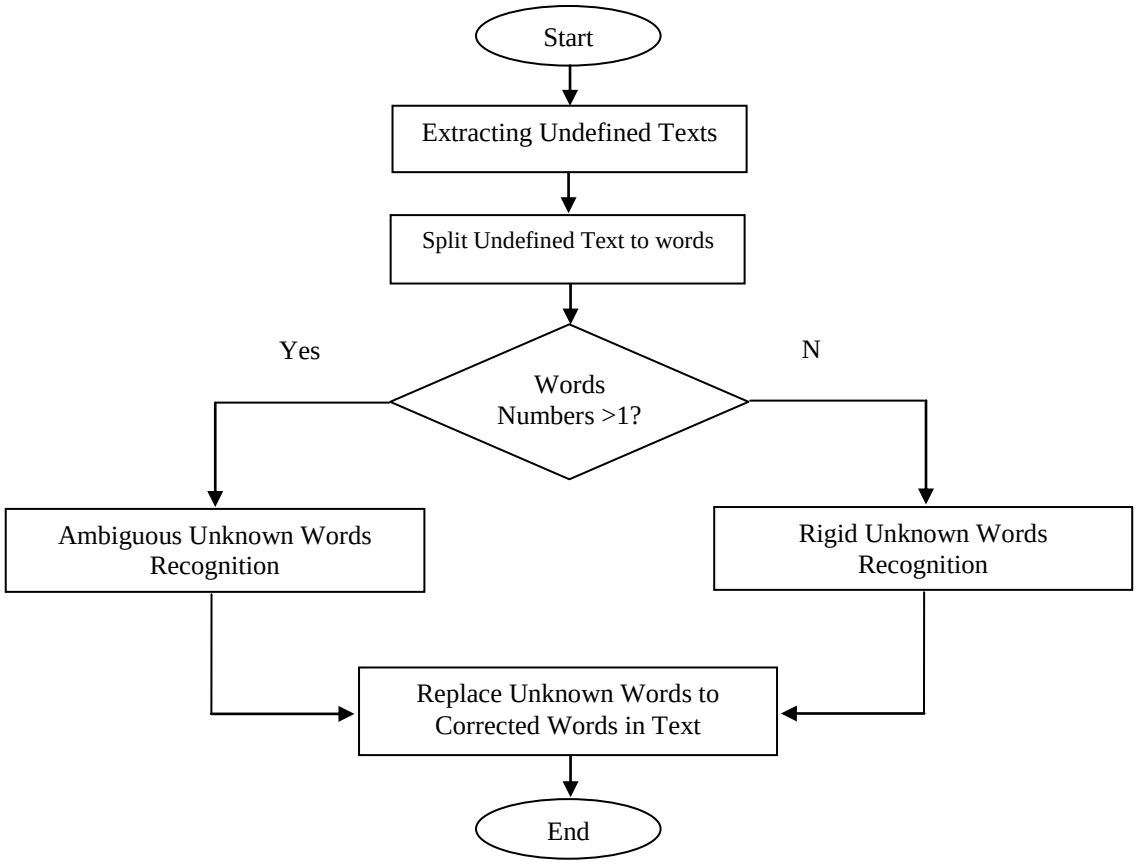


*Figure 4.* The Flowchart of correcting tokenized text

**Correcting Ambiguous unknown words**

After extracting unknown words we split them with"," and a space and produce $UW_s$.

$$UW_s=\{T_1,T_2,...,T_n\}$$
$$T_i=\{ w_1,w_2,...,w_n\}$$

Where $UW_s$ is the set of different terms after splitting unknown text with ',', $T_i$ is one of the produced forms from ambiguous unknown words. Next we split $T_i$ with space; $W_i$ is the word in $T_i$. If all words of $T_i$ be meaningful, then we will process $R_i$ and $F_i$ , $R_i$ refers to term rank and $F_i$ is item frequency. $F_i$ is calculated by equation (1), and $R_i$ is calculated based on the count of meaningful words in $T_i$. The term rank is high if $T_i$ has lower word count. In case of forms with the same rank we select the form that have higher frequency and are more possible. Also we used some rules for meaningful word recognition. In these rules we process word frequency and POS tag.

$$F_i=\{ Fw_1+Fw_2+...+Fw_n\} \quad (1)$$

Where $Fw_i$ is the frequency of $W_i$. For example STep1 cannot detect many special nouns. Such as STep1 produce for "خسروجردی" word two forms like "[خسروجردی، خسرو جر دی]". Since every word in produced form has acceptable POS tag we use frequency feature to detect the correct form. In "خسرو جردی" form, " جر" ("JAR") word is meaningful but in

HAMSHAHRI texts has not been used and its frequency feature value is very lower than threshold. Then "خسرو جر دی" form is not considered and "خسروجردی" saved. We sometimes use POS tag for detects correct form. Namely STep1 produce for "دانش آموزان" (Students) word two forms like "[آموز ان، آموزان]". After processing all words of forms, we delete " آموز ان" form. Because "ان" has not acceptable POS tag and is not an independent word. Table 1 shows more samples that have been corrected by our proposed method.

Table 1
*Some Samples for Corrected Ambiguous unknown words*

| Ambiguous unknown words | Corrected Ambiguous unknown words |
|---|---|
| [خودرا، خو درا، خود را] | خود را |
| [انسان‌ها هستند، انسانهاهستند] | انسان‌ها هستند |
| [عوامل، عوا مل] | عوامل |
| [آنانی، آنا نی] | آنانی |
| [و با اعتماد به، وبا اعتماد به] | و با اعتماد به |
| [خو دبه، خود به] | خود به |

**Correcting Rigid unknown words**

We performed some steps for rigid unknown recognition and correcting them. Figure 5 shows the pseudo code of these steps. We processed rigid unknown words and saved the boundary of meaningful sub words in it. Notice we considered saved boundaries as correct if the last boundary referred to the last character in rigid unknown words. In this method we used some rules for meaningful word correction. Again word frequency and POS tag are important. For example if the extracted sub word of the main word has abbreviation tag (Tag=Ab e.g. "ال", "ان", **...**) then we do not consider it as a meaningful sub word. Also if word frequency is lower than the threshold we do not consider the main word as meaningful.

1- Split unknown words to characters

2- Merge first and second characters and produce sub word

3- If sub word is meaningful, Contain index of last merged character

4- Insert next character

5- If Merge process has been finished and there isn't next character go to 6 else go to 3

6- Add contained index to the list of spaces position

7- If remained string length after space position> 3 go to 8 else go to 9

8- Split the remained string and go to 2

9- Extract last space position from list

10- Extract last sub word by last space position

11- If extracted sub word is meaningful go to 12 else go to 13

12- add spaces to main word in correct positions by spaces list and go to 14

13- write main word without any modification

14- End

*Figure 5.* The Pseudo code of Rigid Unknown Words Correction

Table 2 shows some samples for corrected rigid unknown words.

Table 2

*Some Samples for Corrected Rigid unknown words*

| Rigid Unknown Words | Corrected Rigid Unknown Words |
|---|---|
| [داوطلبصلاحیت] | داوطلب صلاحیت |
| [شمالغرب] | شمال غرب |
| [کلمنته] | کلمنته |
| [انتخابپهلوان] | انتخاب پهلوان |
| [شرایطانشعاب] | شرایط انشعاب |
| [جلبکرد] | جلب کرد |

## Applying Improved Stemmer and POS Tagger

After correcting the bugs of tokenization and producing improved tokenized text, we apply improved stemmer and POS tagger on text. Figure 6 shows our method for removing this problem many times.

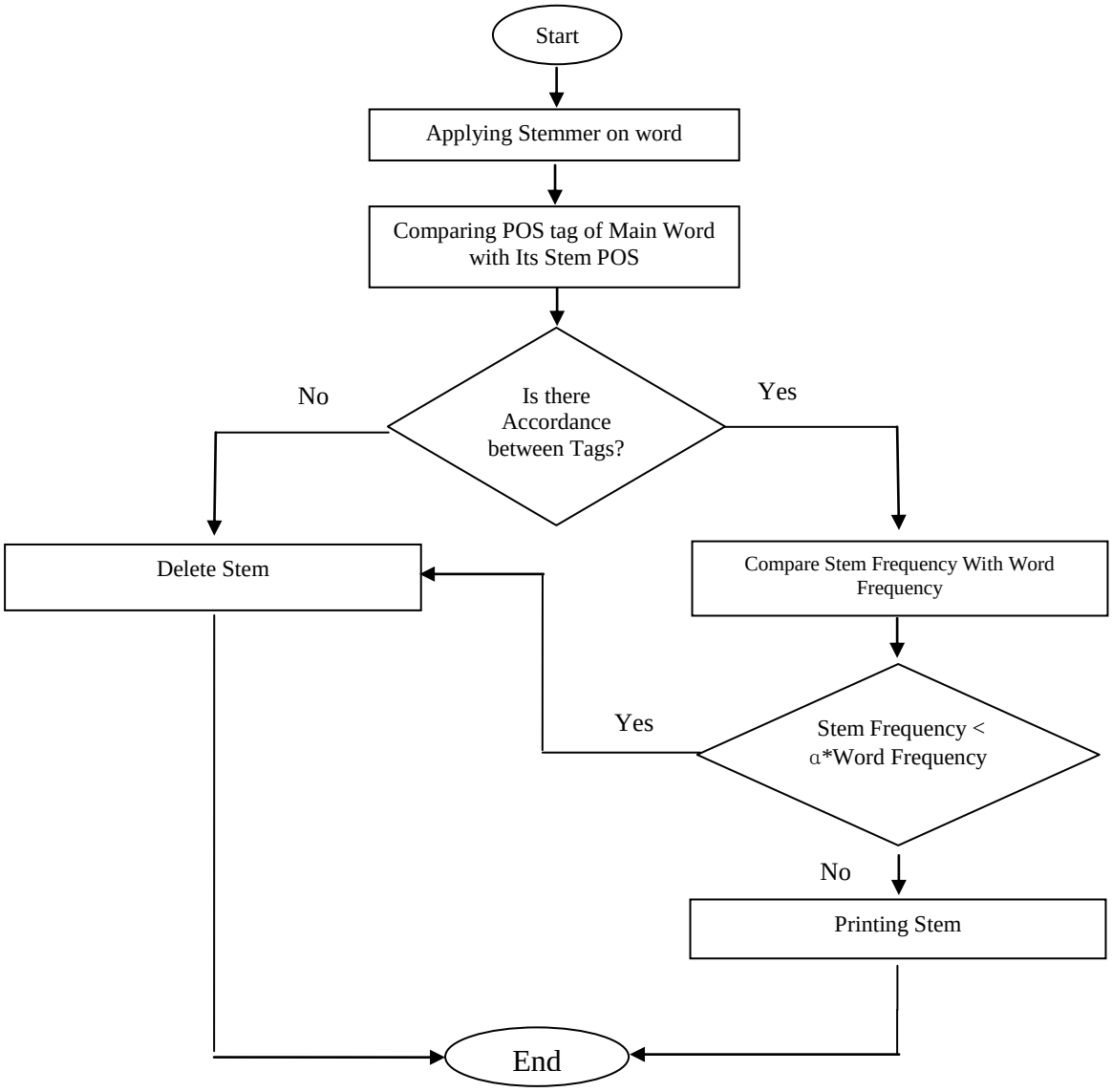


*Fig. 6.* The steps of removing stemmer bug

Firstly we applied STeP-1 stemmer on text. Stemmer detects POS tag, stem, frequency and suffix and prefix of words. This frequency has been calculated based on "bijan khan" data set. Next we improved stemmer result. We compared the POS tag of the main word with POS tag of the word stem according to some rules presented in figure 7. If we found no accordance between tags, we deleted that stem but if there was accordance we compared stem frequency with word frequency. In fact, if stem frequency is not very low, we realized that the stem is normal. For example "ماش" is the stem with very low frequency than "ماشین" therefore this stem is not saved. Notice the best calculated value for α is "0.25". we found this value by some experiments. There are other methods for true stem recognition. We can introduce standard suffix and prefix for each word in lexicon (Eslami et al. 2004) and remove false stems from text but this method is time consuming. Also word sense disambiguation for some words such as homograph words can be useful. Table 3 shows some samples for incorrect stems.

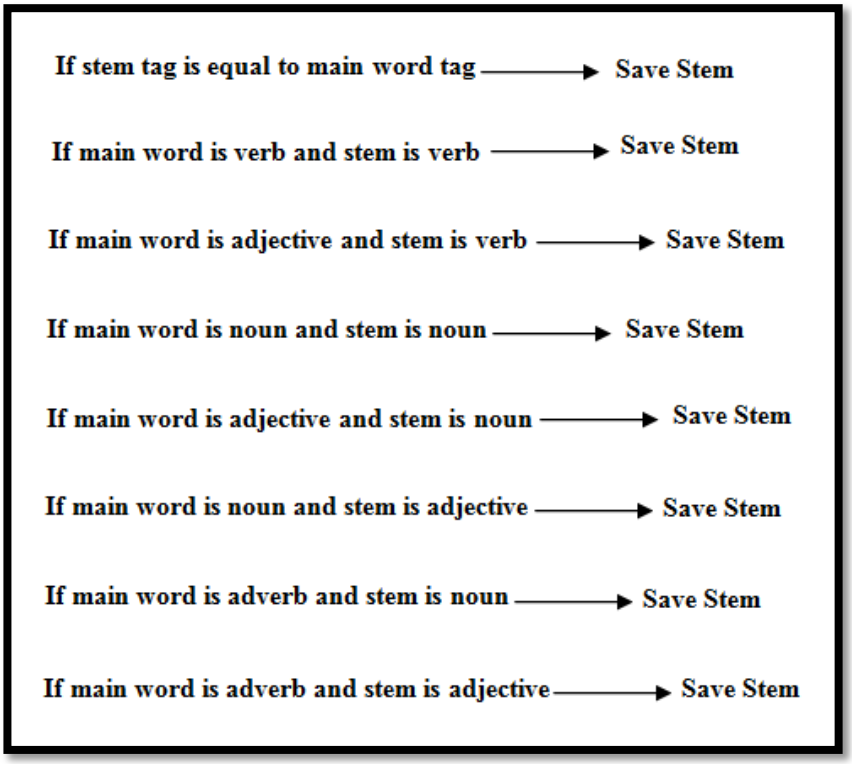


*Figure 7.* The used rules for comparing main word tag and stem tag

Table 3
*Some Samples for Incorrect Stems*

| | | |
|---|---|---|
| جهان (جه) | میلیون‌ها (میلیون,میلی,میل) | بین (ین) |
| همواره (هموار, واره) | است (اس) | بلکه (لکه,بل,لک) |
| گسترش (گستر,گس) | نقوش (نق) | ملکه (ملک,مل) |
| بازدید (زد) | ساده (ساد) | کشورهای (کشور,کش) |

## The Structure of Produced Dataset

The produced corpora are useful for information retrieval and natural language processing. We tried to produce file names of new corpora and some properties similar to the main dataset. Also we added tokenized text, stemmed text, document sentences, words of every sentence, their stems and their POS tags and we saved word stems as well as the main word. The saving form of stemmed text is presented as follows:

$$\text{Main word } (\text{stem}_1, \text{stem}_2 \ldots \text{stem}_n)$$

This saving method enables faster access to the main word and its stem and does not require further processing for stems extraction. Also in information retrieval systems we can search main words and their stems simultaneously. Figure 8 shows the structure of the produced dataset.


*Figure 8.* The structure of produced corpora

## Experiments Results

We selected information retrieval scope for testing the new corpora. Then we used retrieval tools for applying queries in HAMSHAHRI corpora. This helped us test the quality of the new corpora compared with the main corpora. Therefore, we used Lucene.Net which is a standard retrieval tool. Figure 9 shows the results of comparing the performance of the main and produced corpora in an information retrieval system on 100 queries of HAMSHAHRI corpora. We calculated P@5, P@10, P@20 and MAP measures after applying tokenized queries and Stemmed queries on stemmed text and main queries on the main text. The experimental results showed the higher performance of the produced corpus than main dataset. Also we used AND operator between main and tokenized query words and used AND operator and OR between Stemmed Query words. Actually we used OR operator between query words and their stems. Table 4 and figure 9 present experimental results. Also table 5

presents some different applications in information retrieval and natural language processing scopes and describes the influence of the produced corpus on increasing their performance.

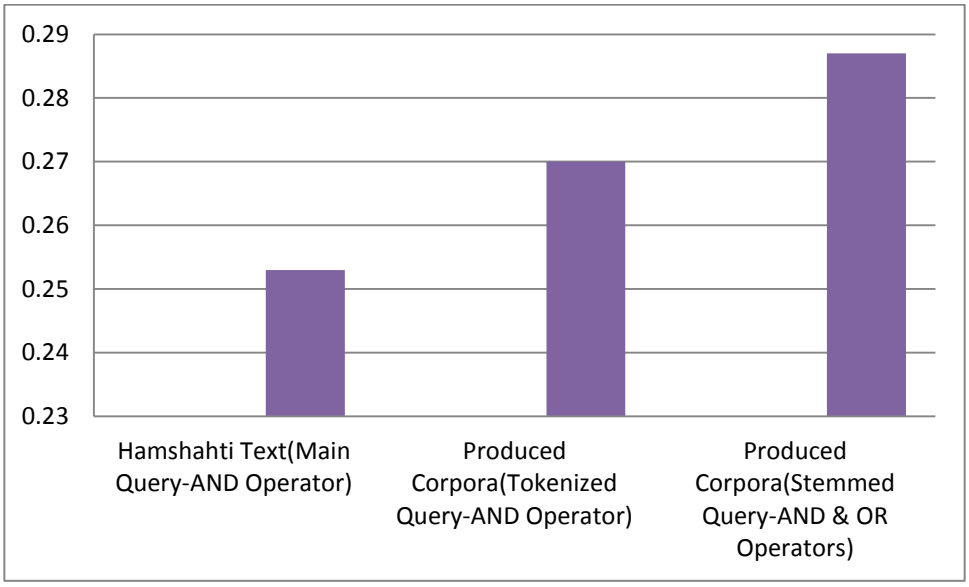


*Figure 9.* The comparing MAP meseaure on main and Produced Corpora

Table 4

*Comparing P@5 and P@10 and P@20 measures on main and produced corpora*

| Text & Query Type | P@5 | P@10 | P@20 |
|---|---|---|---|
| Hamshahri Text (Main Query) | 0.58 | 0.52 | 0.48 |
| HAMTA Text (Tokenized Query) | 0.6 | 0.54 | 0.48 |
| HAMTA Text (Stemmed Query) | 0.61 | 0.55 | 0.51 |

Table 5

*Comparing the influence of produced corpora on increasing applications performance*

| Application Name | The influence of produced corpora on increasing their performance |
|---|---|
| Query Expansion | Since information retrieval is required for implementing many methods in query expansion process therefore better retrieval of data can affect on effective query expansion. Also query expansion doesn't need preprocessing dataset using new dataset. |
| Question and Answering | One step in implementing QA systems is preprocessing which is not required in this system by new dataset. Also extracted sentences and tagged words help to improve the performance. |
| Text Summarization | This application doesn't need to preprocessing dataset using new dataset and extracted sentences and tagged words help to improve the performance. |

| Application Name | The influence of produced corpora on increasing their performance |
|---|---|
| Sense Disambiguation and Conceptual Graph Construction | The tagged words are very effective and concept extraction would be comfortable. Also above notes is included in this application. |
| Named Entity Recognition | Using true written and stemmed and tagged dataset can improve the precision of named entity recognition. |
| Semantic Search Engine Creation | This dataset Helps to concept extraction and retrieval precision improvement. Also This application doesn't need to preprocessing dataset using new dataset. |

## Conclusion

In this research, we focused on creating a suitable dataset for information retrieval and natural language processing in the Persian language. The results showed that the new dataset is effective in improving performance of many applications in Persian information retrieval and Persian natural language processing systems. In future works we will improve our methods for correcting STeP1 bugs and finally we will produce suitable linguistic tools for Persian.

## References

Aleahmad, A., Amiri, H., Rahgozar, M. Oroumchian, F. (2009). Hamshahri: A standard Persian Text Collection. Knowledge-Based Systems, 22(5), 382-387.

Amtrup, J.W., Mansouri Rad, H., Megerdoomian, K., Zajac, R.(2000). Persian-English Machine Translation: An Overview of the Shiraz Project. NMSU, CRL, Memoranda in Computer and Cognitive Science (MCCS-00-319).

Berenjian, S.H. (2013). Persian Simple (Past, Present & Future) Verb Stemmer, Shiraz: Takhte Jamshid. Available At: http://www.ricest.ac.ir. (Persian)

Berenjian, S.H. (2013). The Stemmer of Past and Present from the Infinitive Non Transient Verbs in Persian Language, Shiraz: Navid. Available At: http://www.ricest.ac.ir.(Persian)

Berenjkoob, M. , Mehri, R., Khosravi, H., Nematbakhsh, M.A. (2009). A Method for Stemming and Eliminating Common Words for Persian Text Summarization, Natural Language Processing and Knowledge Engineering, NLP-KE International Conference on, 1-6.

Darrudi E., Baradaran Hashemi, H., AleAhmad, A., Zare Bidoki, A.M., Habibian, A.H., Mahdikhani, F., et al. ( 2008). dorIR collection for Persian web retrieval. Technical Report No. DBRG-TR-02.

Eslami, M., Sharifi, M., Alizadeh, S., Zandi, T. (2004). Persian ZAYA Lexicon. 1st Workshop on Persian Language and Computer, Tehran, Iran.

Estahbanati, S., Javidan, R. (2011). A New Stemmer for Farsi Language. Computer Science and Software Engineering(CSSE), CSI international Symposium on, 3(1), 25-29.

Karimpour, R., Ghorbani, A., Pishdad, A., Mohtarami, M., Aleahmad, A., Amiri, H., Oroumchian, F. (2009). Improving Persian information retrieval system using stemming and part of speech tagging. 9th workshop of the cross-language evaluation forum.

Mehrad, j., Berenjian, S.R. (2011). provided a Persian language singular stemmer system, International Journal of Information Science and Management, 9(2).

Miangah, T.M. (2009). Constructing a Large-Scale English-Persian Parallel Corpus. Meta:Translators' Journal 54(1), 181-188.

Oroumchian, F., Tasharofi, S., Amiri, H., Hojjat, H., Raja, F.(2006). Creating a Feasible Corpus for Persian POS Tagging. UOWD Technical Reports Series , Number TR 3.

Oroumchian, F., Darrudi, E., Hejazi, M.R. (2004). Assessment of a modern Persian corpus. Proceedings of The 2nd Workshop on Information Technology & its Disciplines (WITID), ITRC, Iran.

Rashidi, A., Zolfy Lighvan, M. (2014). HPS: A Hierarchical Persian Stemming Method. International Journal on Natural Language Computing (IJNLC), 3(1).

Rezvan, Y., Ghandchi, M., Rezvan, F. (2009). Suggesting Correct Words Algorithms Developing in FarsiTeX. Proceedings of the European Computing Conference.

Shamsfard, M., Jafari, H.S., Ilbeygi, M. (2010). STeP-1: A Set of Fundamental Tools for Persian Text Processing. LREC 2010 - 8th Language Resources and Evaluation Conference, 19-21 May, Malta.

Sheykholeslam, M.H., Minaei-Bidgoli, B., Juzi, H. (2012). A Framework for Spelling Correction in Persian Language Using Noisy Channel Model. In Proceedings of Language Resources and Evaluation Conference (LREC).

Sheykh Esmaili, K., Abolhassani, H., Neshati, M., Behrangi, E., Rostami, A., Mohammadi, M. (2007). Mahak: A Test Collection for Evaluation of Farsi Information Retrieval Systems. IEEE/ACS International Conference on Computer Systems and Applications.

Taghiyareh, F., Darrudi, E., Oroumchian, F., Angoshtari, N.(2003). Compression of Persian Text for Web-Based Applications, Without Explicit Decompression. WSEAS Transactions on Computers, 2(4), 961-966.

Talvensaari, T., Pirkola, A., Jãrvelin, K., Juhola, M., Laurikkala, J. (2008). Focused web crawling in the acquisition of comparable corpora. Information Retrieval 11, 427- 445.

Tashakori, M., Meybodi, M.R., Oroumchian, F. (2002). Bon: The Persian Stemmer. Information and Communication Technology - EurAsia-ICT , 487-494.

Yang, C.C., Li, K.W. (2004). Building parallel corpora by automatic title alignment using length-based and text-based approaches. Information Processing & Management 40(6), 939-955.