A Semi-supervised Approach for Key-Synset Extraction to Be Used in Word Sense Disambiguation

Maryam Haghollahi and Mehrnoush Shamsfard

NLP Research Lab, Faculty of ECE, Shahid Beheshti University, Tehran, Iran
maryam.haghollahi@yahoo.com, m-shams@sbu.ac.ir

Abstract. Nowadays, although many researches is being done in the field of word sense disambiguation in some languages like English, still some other languages like Persian have many things to be done. Some difficulties are in this way which might have made it less interactive for researchers. For example, Persian WordNet or FarsNet is newly developed and there is no sense tagged corpus developed based on it yet. So we propose a semi-supervised approach for extending FarsNet with some new relations and then use it for WSD. Also a method to extract semantic keywords or key-concepts from textual documents is used. As the key-concepts are extracted exploiting FarsNet, we call them Key-synsets. In fact Key-synsets of a document are those synsets which are semantically related to the main subjects of that document. This method is exploited to improve the precision of the proposed WSD. Although our approach is tested on Persian it can be easily adopted for other languages such as English.

1 Introduction

Word sense disambiguation is a critical task in many applications like translation and semantic search. Many approaches are implemented in order to facilitate it; some of them are supervised and some others are unsupervised. In English some sense tagged corpora which are tagged with WordNet can be found, like SemCor [1]. Thus supervised approaches have the feasibility to be implemented. But in some languages like Persian, there isn't any sense tagged corpus to be used for the learning phase.

In this paper, we introduce a software system which extracts the candidate senses (and so synsets) of the words within a context using FarsNet relations; and then with respect to the relations between these candidate synsets, it finds the Key-synsets of the context to make the approach more precise. In this way, some new relations are extracted semi-automatically and added to FarsNet. The results show a significant improvement in precision with these new relations.

2 Related Work

This paper discusses a key concept extraction method to be used in word sense disambiguation. Thus in this section we briefly point to the related work on WSD and keyword extraction.

M.V.M. Salem et al. (Eds.): AIRS 2011, LNCS 7097, pp. 594–603, 2011.

According to [2] word sense disambiguation methods are categorized into three categories; supervised, unsupervised and knowledge based. Some other researchers have different categorizations. Tsatsaronis and colleagues [3] categorize WSD methods into supervised and unsupervised and say that unsupervised WSD methods comprise corpus-based [4], knowledge-based such as Lesk-like [5] and graph-based [6] methods, as well as ensembles [7] that combine several methods.

Supervised approaches (such as [8]) use corpora which are tagged with the concepts of ontologies, for their training phase. While semi-supervised approaches restrict the need to such a resource. For instance Tang and colleagues [9] use the examples of an ontology and some raw text resources and extract their subject-predicates and predicate-objects as collocation words. Then the training data are obtained with two approaches: SRP which means that within all possible senses sets of a word collocation, the one which has the most redundant information between senses is the best; and PRP which is calculated by exchanging the words with their synonyms, etc. and finding the most co-occurring ones.

Unsupervised approaches often are used for languages which have no or less concept tagged corpus for training phase. Results show that supervised approaches are often more precise but limited to those words that have sense tagged data [10 quoting [11]. For example, Tsatsaronis and colleagues [12] utilize WordNet, and use neural network and a spread activation method to disambiguate the words of sentences. Tran and colleagues [13] construct a wide tree of word relations with their weights using many internet web pages. Then for disambiguation of a word in a context, the glosses of all of its senses and the context of the to-be-disambiguated word are parsed and using the constructed tree each gloss obtains a score. The sense whose gloss obtains the most score will be selected for that word.

Knowledge based approaches use some knowledge resources like dictionaries or thesauri for WSD task. As Navigli [2] mentioned, their precision is less than supervised methods but their coverage is more expanded because of the expanded resources they have. Lesk [5] and extended Lesk [14] are two of these methods. They use the glosses of the senses in WordNet to disambiguate the words.

Recent research results [2] show that “the accuracy of the state of the art supervised WSD methods is above 60% with an upper bound reaching 70% for all words, fine-grained WSD for English, while the accuracy of unsupervised methods is usually between 45 – 60%”. There are some known Baselines which can be used for evaluation phase of WSD works. The best known of them is the First-Sense approach. Also Lesk method can be used as a baseline, as Navigli [2] mentioned.

In Persian, as there is no corpus tagged by word senses, there is no supervised work on WSD. Saedi and Shamsfard [15] propose a knowledge based WSD method to be used in a Persian to English machine translation system. Faili [16] introduces an English to Persian translation method which has a WSD approach on English texts that uses a parallel corpus for its training phase. There is no work which assigns senses according to a Persian WordNet so far.

Keyword extraction is another field related to the subject of this paper. Many features are used for keyword extraction process. For example Xu and colleagues [17] use Wikipedia to derive a set of novel word features which reflect the document’s background knowledge. These features are the inlink, outlink, category and infobox information of the document’s related articles in Wikipedia. Ercan and colleagues

[18] concern the relations between the words of the document to extract the keywords. In fact, using a supervised method, a lexical chain is developed from each document by using WordNet to be used for keyword extraction.

Hulth [19] explains some methods for extracting the keywords. Some methods use the syntactic information of words (such as [20]), some have supervised learning phase (such as [21]) and some others are statistical (such as [22]).

3 Word Sense Disambiguation Approach

The proposed method is composed of three essential processes: Stemming and tokenizing, Word sense disambiguation and Key-synset extraction. Stemming and tokenization of Persian documents are done by STeP-1 software [23]. In the rest of this section, we will discuss WSD and Key-synset extraction approaches in more details.

3.1 Persian Word Sense Disambiguation

FarsNet [24] is a Persian WordNet, recently developed in NLP lab of Shahid Beheshti University. In this lexical ontology, various kinds of relations are defined between synsets including: Hypernym, Hyponym, Meronym, Holonym, Antonym and Cause. Many researchers have used these kinds of relations to disambiguate the words senses. For example Fragos et al. [25] proposed a method to find the words senses using WordNet relations. Here, we have used synsets' relations of FarsNet to find the senses of words.

Our experiments showed that the above relations are not enough to find the word senses, because some combinations of related words haven't got any of these kinds of relations. For instance, human's mind comprehends a semantic relation between "شیر" (shir, means: lion) and "جنگل" (jangal, means: jungle). But this relation is not among the above kinds. We call these relations just as "Is related to" without putting any specific name or label on them. Neither FarsNet nor most of the other WordNets include this kind of relation. We have extended FarsNet relations with a few new "Is related to" relations for some concepts semi-automatically. Results of using these new relations for disambiguation showed that they really increase the precision.

Semi-automatic extraction of semantic relations. As it was described before, extracting semantic relations between synsets can improve the precision of search. Here, we have used a semi-automatic approach to do it. To find the words which are related to a target word, first we search it via Hamshahri-1 corpus [26] with tf-idf method and retrieve some highest ranked documents. In each document, we extract the words within a 5-words sized window around the target word which are not stop words. Then, all possible synsets of these adjacent words and their hypernyms up to two levels are extracted from FarsNet and added to a list. The frequency of occurrence of each of these synsets in the obtained list is considered as its rank. "n" best synsets with respect to their ranks are semantically related synsets with the target word. It should be considered that the synset of the target word is assigned manually, but the synsets of its co-occurring words are obtained automatically. Thus the approach is semi-automatic.

Some of the co-occurring words of our target words are not presented in FarsNet and though with this approach we will lose them. Thus, we introduce a new relation type which is between a synset and an unkown word (word which is not in FarsNet). It means that although we haven't got that word in FarsNet and don't know which synset it can occurs in, but we know that this word is co-occurring with some specific sense of the target word. These relations will be considered as direct relations in FarsNet in the process of disambiguation. Our results show some improvements in precision by adding these new relations.

Using FarsNet for ranking the synsets' semantic relations. Having FarsNet relations, we can find the weight of the relation between any two FarsNet synsets. Equation (1) shows how to calculate this weight. In this equation distance(S_i,S_j) is the number of relations that should be passed from S_i to arrive to S_j.

$$\text{Weight}(S_i\ ,\ S_j) = \log_{10}\left(\frac{1}{\text{distance}(S_i,S_j)}\right) + c. \qquad (1)$$

In the next part we will explain the use of this weighting process for our disambiguation algorithm.

Finding the set of words' synsets of each block of content. To disambiguate the words of each document, we need to split it to smaller blocks. Then, in each block we can find the relatedness weight of any two synsets of any two words to find the best synsets of the block's words. In this work, we examined some different number of words within blocks to know which one is better. The results and comparisons are presented in our results section. Also as it is mentioned in our results section, the jump number of blocks has an important effect on the efficiency of the approach. Jump number is the number of words that we will pass after disambiguation of a block, in order to obtain the next block. For example, if we set the block's number of words to 4 words and set the jump number to 2, the sentence "What is the past tense of split?" will be split to these three blocks (here without omission of stop words):

"What is the past", "the past tense of" and "tense of split - ".

Having each block of "n" words, each possible sense (and so synset) of each word is found from FarsNet, and then the relatedness weight of any two synsets of them will be calculated as described before. Here, we only consider direct relations of synsets and ignore indirect ones to reduce the computation time. Now we have a set of synset pairs with their similarity weights. Equation (2) shows how to calculate the total score of each pair. In this equation, pos(W_j) and pos(W_i) are respectively the position of second and first words in the current block.

$$\text{TotalScore(i,j)} = \text{coef(i,j)} * \text{Weight(i,j)}. \qquad (2)$$

$$\text{coef}(i,j) = \log_{10}\left(\frac{1}{\text{pos}(W_j)\ \text{-}\ \text{pos}(W_i)}\right) + c.$$

After acquiring the set of pair synsets with their total scores, each two pairs which are not mutex will be merged together to build a bigger set. Two pairs are mutex if they contain at least a similar word with different senses. After merging the pairs, the score of the new collection is the addition of the scores of its components. This process will be continued until no new collection can be added. After that, the collections will be sorted by their scores, and better collections are extracted for the next phase.

3.2 Key-Synset Extraction Approach

The main idea of this method of key synset extraction is taken from a general principle about information density. For example, in clustering algorithms, the points which are inside a dense part have more probability to be a cluster, and points which have less density may be noises.

Inspired by this idea, we claim that those senses of a document which have more valuable relations with other senses are more probable to be key senses. In continuation of this section, this method will be described in more details.

Calculating the pseudo-frequency. As we described in previous sections, better combinations of synsets of each block of "n" words will be used to find the key synsets. Actually, each synset which occurs in any of the best combinations can be a candidate to be a key synset. So, we will calculate a rank for each of them to find better ones. To calculate this rank, first using equation (3) we will compute a pseudo-frequency for each synset, which *somehow* shows the amount of its occurrences in the document. In this equation, the pseudo-frequency of i^{th} synset is calculated. $ColSet_i$ is the set of collections that i^{th} synset occurs in them and $Score(Col_k)$ is the score of k^{th} collection.

$$SF_i = \sum\nolimits_{k \in ColSet_i} Score(Col_k). \tag{3}$$

Calculating the total ranks. The last thing we need is to compute the total rank of each synset in each document, which is the main criterion to find the Key-synsets. What we need, is to calculate the relation score between each two candidate synsets of the document as described before. Here, we used indirect relations with the threshold of "at most ten fathers" in addition to direct ones for the process of relation scoring. We used this parameter (relation score) before for some other reasons. Here we are using it to calculate the importance of each synset in the document, but before, we used it to disambiguate the senses of the words of "n" words blocks.

Finally we have everything we need! Using Equation (4) we can calculate the total rank of each synset to find Key-synsets of each document's words. Here, "k" is the number of candidate synsets.

$$TR_i = SF_i * \sum\nolimits_{j=1:k} SF_j * Weight(S_i , S_j). \tag{4}$$

4 Experimental Results

As there is no Persian sense tagged corpus being tagged with FarsNet, we had to make our own test corpus. We have used Hamshahri-1 corpus for this task. First of all, we divided the corpus into %70 training and %30 testing corpora randomly. Using these corpora, we can find the new relations and evaluate our approach.

4.1 Training Phase

First we chose three ambiguous words that more than one of their senses occur in the corpus. These words are “شیر” (“shir” means: lion, faucet, breastfeeding, milk), “سیر” (“sir” means: garlic, full. or “seir” means: process, travel) and “گل” (“gol” means: flower, goal). Then these three words have been searched within training corpus with tf-idf measure. For our search, we used Lucene[1] search engine that uses tf-idf measure in order to rank the results of search. Within the highest ranked documents, we extracted the co-occurring words. Then with the explained method, new relations between synsets and between unknown words and synsets were extracted.

4.2 Building the Test Corpus

In order to build the test corpus, those three words were searched within test corpus and about 600 “nearly 200 character phrases” around our specified words were extracted from the retrieved documents to build a test corpus. Our three words were tagged manually within the test set and the content of each phrase was stemmed with STep-1. Then the proposed method was evaluated via this built corpus over the tagged words.

4.3 Evaluation of the Method

The proposed method was tested over the built test corpus with different states. Each state will be described below and its results will be presented. The programs are written in java and the tests are being done on an ordinary PC with 2GB RAM and 2.66GHz CPU.

Precision of baselines. Lesk and extended Lesk approaches were implemented as our baselines, because there was no Persian WSD method working with FarsNet to be compared with our method. Also as there were no sense tagged corpora, the precision of MFS couldn’t be calculated, but here, we calculated the best case of MFS within the test corpus and used it as another baseline. Table 1 shows the precision of baselines.

[1] http://lucene.apache.org/

Table 1. Precision of baselines

Method	Precision
First Sense (In best case)	64%
Lesk	10%
Extended Lesk	16%

Precision of our approach with different parameters and features. The proposed approach has some kinds of parameters and features. Results show that changing them have a significant effect over the precision. Here, first we will show the effect of these parameters, then we will compare our approach with baselines and eventually we will show the effect of features.

Word number and jump number effect. We have calculated the precision of our approach over the test corpus with different words number of blocks and jump number. Fig. 1 and Fig. 2 show the experimental results.

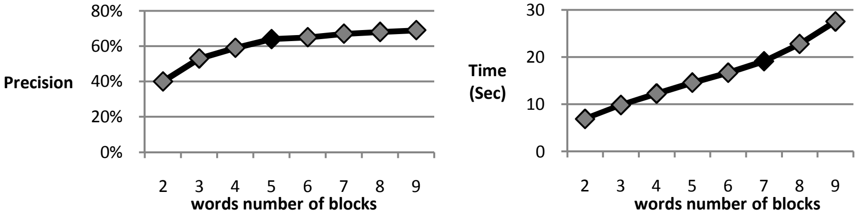

Fig. 1. The left curve shows the Precision of proposed approach with different words number of blocks and jump number of one. The right curve shows the computation time of each complete phrase (each "nearly 200 character phrase") in test set with mentioned conditions.

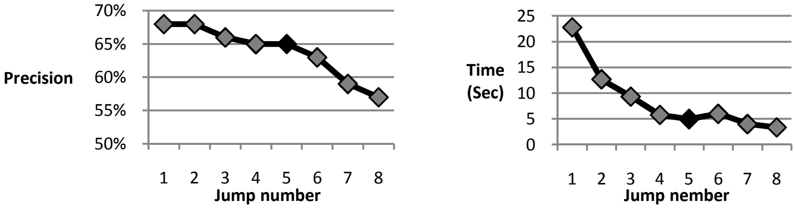

Fig. 2. Left curve shows the precision of proposed approach with 8 words in each block and different jump numbers. Right curve shows the computation time for each phrase with mentioned conditions.

Fig. 3 shows the comparison of the precision of our approach with the base lines. Our approach's precision is calculated with different words number of blocks and jump number of one. Results show that our approach has out-performed the baselines.

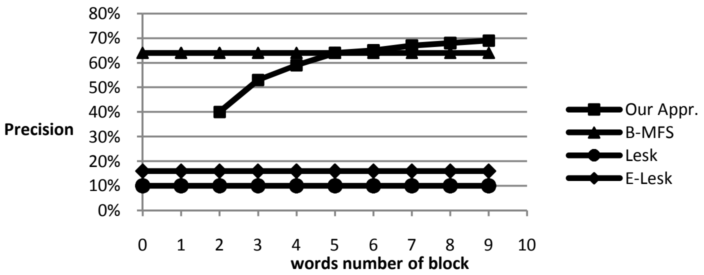

Fig. 3. comparison of our approach with baselines. B-MFS stands for Best case of MSF and E-Lesk stands for Extended Lesk.

Different features effect. We have calculated the precision of our approach with and without each of the proposed features. Table 2 shows the results of the test.

Table 2. Precision of our approach with different features. Here "+ something" means that the state has that thing and "- something" means that it doesn't have it. Also syn-syn relation means the added relations which are between two synsets and unk-syn relations are the added relations that are between an unknown word and a synset. Our approach's precision is calculated with concerning 4 words inside each block and jump number of 2 as an example. As it can be understood from this table, without the extraction of new relations for FarsNet, this approach is so inefficient, but with adding new relations, the precision grows much. Also adding the relations between unknown words and synsets will improve the method by about 8 percent. In addition, without the post-process of Key-synset extraction, the results show a worse precision.

Feature			**Precision**
- syn-syn rels.	**- unk-syn rels.**		7%
+ syn-syn rels.	**- unk-syn rels.**		48%
- syn-syn rels.	**+ unk-syn rels.**		23%
+ all other features			55%
+ syn-syn rels.	**+ unk-syn rels.**	**- Key-synset**	44%

Complexity of the method. If we assume that the words number of block is "n", the average senses of each word is "m", the jump number is "j" and the average number of words in each context is "c", then the complexity of finding Key-synsets of each document in its worst case is calculated in equation (5). In this equation $\frac{c}{j}$ is the number of blocks in the document and $\binom{m.n}{2}*\binom{m.n}{2}*\binom{m.n}{2}$ is the complexity of disambiguating each block.

$$O\left(\frac{c}{j}*\binom{m.n}{2}*\binom{m.n}{2}*\binom{m.n}{2}\right)= O\left(\frac{c * m^6 * n^6}{j}\right) . \tag{5}$$

5 Conclusion and Future Works

We have presented an approach for WSD with FarsNet on Persian texts. Our approach uses both FarsNet relations and some other relations that are extracted by a semi-supervised approach and added to FarsNet. In addition, we made an automatic overall revise over the detected synsets of words to make them more precise, or in other words, we found the Key-synsets of the distinct words within the context. The results show improvement in the precision with adding these new features. Also our approach out-performs First sense, Lesk and extended Lesk with respect to the experimental results. The most important disadvantage of our approach is its computation time which can be better with making some optimizations in approach. Of course it should be considered that the computation times are calculated within our runs over an ordinary PC and they would be much better over a stronger server.

For our future work we might optimize our approach to make it faster. Then we will develop a semantic search engine to use this approach for indexing the documents. We think that common search engines don't work well in Persian and have no sense to the semantic of queries and documents. Thus, we are going to apply these semantic methods over search engines to make the results more admissible.

References

1. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: Proc. of the ARPA Workshop on Human Language Technology, pp. 303–308 (1993)
2. Navigli, R.: Word Sense Disambiguation: A Survey. ACM Computing Surveys 41(2), Article 10 (February 2009)
3. Tsatsaronis, G., Varlamis, I., Nørvåg, K.: An Experimental Study on Unsupervised Graph-based Word Sense Disambiguation. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 184–198. Springer, Heidelberg (2010)
4. Yarowsky, D.: Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In: Proc. of COLING, pp. 454–460 (1992)
5. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. of the 5th SIGDOC, pp. 24–26 (1986)
6. Agirre, E., Soroa, A.: Personalizing pagerank for word sense disambiguation. In: Proc. of EACL, pp. 33–41 (2009)
7. Brody, S., Navigli, R., Lapata, M.: Ensemble methods for unsupervised wsd. In: Proc. of COLING/ACL, pp. 97–104 (2006)
8. Agirre, E., Martínez, D.: Exploring automatic word sense disambiguation with decision lists and the Web CoRR cs.CL/0010024 (2000)
9. Tang, X., Chen, X., Qu, W., Yu, S.: Semi-Supervised WSD in Selectional Preferences with Semantic Redundancy. In: COLING (Posters), pp. 1238–1246 (2010)
10. Brody, S.: Closing the Gap in WSD: Supervised Results with Unsupervised Methods. Doctor of Philosophy thesis, Institute for Communicating and Collaborative Systems. School of Informatics. University of Edinburgh (2009)
11. Yarowsky, D., Radu, F.: Evaluating sense disambiguation across diverse parameter spaces. Natural Language Engineering 9(4), 293–310 (2002)
12. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. In: Proc. of IJCAI, pp. 1725–1730 (2007)
13. Tran, A., Bowes, C., Brown, D., Chen, P., Choly, M., Ding, W.: TreeMatch: A Fully Unsupervised WSD System Using Dependency Knowledge on a Specific Domain. In: Proc. of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden, July 15-16, pp. 396–401 (2010)

14. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proc. of the 18th International Joint Conference on Artificial Intelligence, IJCAI, Acapulco, Mexico, pp. 805–810 (2003)
15. Saedi, C., Shamsfard, M.: Translating Persian documents into English using knowledge based WSD. In: ICDIM 2009, pp. 229–234 (2009)
16. Faili, H.: An Experiment of Word Sense Disambiguation in a Machine Translation System. In: Proc. of 2008 IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2008), pp. 28–35 (2008)
17. Xu, S., Yang, S., Lau, F.: Keyword Extraction and Headline Generation Using Novel Word Features. In: Proc. of the Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
18. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. Information Processing and Management 43(6), 1705–1714 (2007)
19. Hulth, A.: Automatic Keyword Extraction. VDM Verlag Dr. Mueller, E.K. Binding: Paperback (2008) ISBN: 363903855X, ISBN-13: 9783639038552
20. Barker, K., Cornacchia, N.: Using Noun Phrase Heads to Extract Document Keyphrases. In: Canadian Conference on AI 2000, pp. 40–52 (2000)
21. Turney, P.D.: Learning algorithms for keyphrase extraction. Information Retrieval 2(4), 303–336 (2000) (NRC #44105)
22. Wartena, C., Brussee, R., Slakhorst, W.: Keyword Extraction Using Word Co-occurrence. In: Workshops on Database and Expert Systems Applications, Bilbao, Spain, August 30-September 03 (2010) ISBN: 978-0-7695-4174-7
23. Shamsfard, M., Jafari, H., Ilbeygi, M.: STeP-1: A Set of Fundamental Tools for Persian Text Processing. In: LREC (2010)
24. Shamsfard, M., Hesabi, A., Fadaei, H., Mansoory, N., Famian, A., Bagherbeigi, S., Fekri, E., Monshizadeh, M., Assi: Semi Automatic Development of FarsNet; The Persian WordNet. In: Assi: Semi Automatic Development of FarsNet; The Persian WordNet. 5th Global WordNet Conference (GWA 2010), Mumbai, India (2010)
25. Fragos, K., Maistros, I., Skourlas, C.: Word Sense Disambiguation using WordNet relations. In: Proc. of 1st Balkan Conference in Informatics, October 20-22 (2003)
26. AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., Oroumchian, F.: Hamshahri: A standard Persian text collection. Journal of Knowledge-Based Systems 22(5), 382–387 (2009)