# Unsupervised Word Sense Disambiguation Using Word Embeddings

Behzad Moradi*, Ebrahim Ansari*†, and Zdeněk Žabokrtský†
*Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran
{behzadmoradi, ansari}@iasbs.ac.ir
†Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic
{ansari, zabokrtsky}@ufal.mff.cuni.cz

***Abstract*—Word sense disambiguation is the task of assigning the correct sense of a polysemous word in the context in which it appears. In recent years, word embeddings have been applied successfully to many NLP tasks. Thanks to their ability to capture distributional semantics, more recent attention have been focused on utilizing word embeddings to disambiguate words. In this paper, a novel unsupervised method is proposed to disambiguate words from the first language by deploying a trained word embeddings model of the second language using only a bilingual dictionary. While the translated words are useful clues for the disambiguation process, the main idea of this work is to use the information provided by English-translated surrounding words to disambiguate Persian words using trained English word2vec; well-known word embeddings model. Each translation of the polysemous word is compared against word embeddings of translated surrounding words to calculate word similarity scores and the most similar word to vectors of translated surrounding words is selected as the correct translation. This method only requires a raw corpus and a bilingual dictionary to disambiguate the word under question. The experimental results on a manually-created test dataset demonstrate the accuracy of the proposed method.**

## I. Introduction

Word Sense Disambiguation (WSD) is a key task in the field of natural language processing. Some words in spoken languages have more than one meaning and they may exhibit different senses depending on which context they appear; and the aim of WSD is to disambiguate these senses automatically. WSD is not considered as an end task in NLP [1] and is essential for other tasks such as machine translation [2], [3], [4], information retrieval [5], question answering [6], sentiment analysis [7], and text summarization [8]. In order to resolve the semantic ambiguity, many approaches have been proposed so far [9], [10], [11], [12], [13], [14], [15]. However, we can categorize these methods into three main methodologies: Knowledge-based, supervised and unsupervised systems [16]. Knowledge-based methods need some external resources such as machine readable dictionaries, thesauri, lexical knowledge bases, etc. Although the supervised methods often outperform the unsupervised ones, they need a large quantity of hand-tagged data.

This paper proposes a new unsupervised WSD method using word embeddings and a small dictionary. In contrast to most existing techniques, which resolve semantic ambiguity of a target language using the information provided by that language, we exploit rich resources from another language to overcome the problem. Based on the fact that contextual information provides useful clues for disambiguating words, we introduce an approach to resolve semantic ambiguity which uses the translation of the surrounding words of the Persian ambiguous word into English. To select the correct sense of the ambiguous word, its candidate senses (here translations) should be compared with English-translated surrounding words. The process of finding similarity between translations of the ambiguous word and the translated surrounding words is done inside the embedding space instead of using classic co-occurrence vectors. Therefore, we build our word embedding model from English Wikipedia articles [17].

In this work, we selected Persian for our WSD process. Persian (Farsi) is a member of the Indo-European language family within the Indo-Iranian branch and is spoken in Iran, Afghanistan, Tajikistan and some other regions related to ancient Persian. The Persian alphabet has 26 consonants and six vowels. However, three short vowels rarely are shown in writing except for beginner learners. The absence of these short vowels makes lexical ambiguity in some words, e.g., words "گل" 'flower' [Gol] or 'mud' [Gel], "جو" 'atmosphere' [Jav] or 'barely' [Jo] and "سبک" 'light-weight' [Sabok] or 'style' [Sabk].

This paper is structured as follows. Section II gives an overview of methodologies and some related work and also a brief history and description of word embeddings is discussed. Our proposed method is introduced in Section III. Section IV illustrates experimental setup including dataset and model training, as well as results and comparison against other works. Finally, Section V concludes the paper.

## II. Related work

Our main goal is to design an unsupervised approach for WSD by leveraging word embeddings. In Section II. I, an overview of WSD methodologies and some related work are reviewed and Section II. II contains a brief description of word embeddings and its application in WSD.

### A. Word sense disambiguation

Several approaches to word sense disambiguation has been proposed in recent years [9], [10], [11], [12], [13], [14], [15]. These approaches can be categorized as three main methods: Knowledge-based, Supervised and Unsupervised. Knowledge-based approaches exploit external knowledge resources, such as Machine Readable Dictionaries (MRD), thesauri, lexical knowledge bases, etc. Generally, the main idea behind the

ISSN 2305-7254

supervised methods is that a context can provide enough information to create a model to disambiguate words. Thus, supervised methods need sense-annotated corpora in the training phase. On the other hand, unsupervised methods do not require manually labeled data to disambiguate words. Because of covering a broad range of domains and being language-independent, unsupervised WSD seem appealing to many researchers. As remarkable work on WSD, we can mention the method proposed by Banerjee and Pedersen (2002). They modified Lesk [9] to benefit from advantageous information of WordNet [18] hierarchical relations [12]. Given some ambiguous words, the Lesk (1986) algorithm compares the definition of each sense of a word against the definitions of every other words. Then a word is assigned a meaning which its description shares the largest number of common words with the definitions of other words of the sentence. While the Lesk algorithm considers only the definition of a word sense by looking up traditional dictionaries, Banerjee and Pedersen enlarge contextual information using WordNet synsets. Also, while the ordinary Lesk algorithm compares only the definitions of words being disambiguated, the adapted Lesk algorithm is able to compare the definitions of the surrounding words. Among unsupervised WSD methods, some use parallel or comparable corpora to resolve the problem. For example, Kaji and Morimoto presented an unsupervised method for WSD using bilingual comparable corpora [13]. First, they extracted pairs of related words from the corpus of each language. Then, after aligning these pairs of related words, translingual correlation between senses of the ambiguous word and the words related to the ambiguous word is calculated and finally, they select the sense that maximizes sum of correlations [13].

### B. *Word embeddings in WSD*

The traditional word representations, for example, one-hot representation treat words as discrete and unique entities. In such a setting, information about one word gives no useful clues about other co-occurrent words in the context. Representing words in this way furthermore leads to data sparsity. Unlike one-hot representations, distributed representations, i.e., word embeddings, capture contextual information to represent words in a multi-dimensional space. Word embeddings are feed-forward neural networks that take words as input and map them into low-dimensional vectors. These vectors contain useful information about both semantic and syntactic relationships between the target word and other words. More precisely, a vector represents each word and the words are mapped into a continuous vector space, where semantically related words have similar vector representations, some basic vector operations on vectors are applicable.

The word "embedding" was originally introduced by Bengio et al. (2003). Their neural language model predicts the next word in the sentence, given previous words using a feed-forward neural network having one hidden layer. Collobert and Weston (2008) introduced a general deep neural network architecture for NLP and applied their model to some tasks such as semantic role-labeling, part-of-speech-tags, chunks, named-entity recognition and language modeling. However, Mikolov et al. (2013) did the most promising work in 2013. They introduced word2vec, a computationally efficient toolkit to represent words as vectors. To embed words, word2vec uses two architectures: the continuous bag of words (CBOW) and skip-gram. CBOW predicts a target word given its context; it uses $n$ words before and after target word to predict target word. The second model, skip-gram is the reverse of CBOW, i.e., predicts the context words given the target word. According to Mikolov et al. (2013), the training of CBOW is faster than skip-gram and has better accuracy for frequent words, but skip-gram is suitable for a small amount of training data, and represents infrequent words well. An algorithm similar to word2vec was presented by Pennington et al. (2014), named Glove. Regular neural networks often produce task-specific embeddings, but in comparison, the embeddings produced by word2vec and Glove are useful in many downstream tasks.

In recent years, there has been a considerable amount of research on utilizing word embeddings in WSD [14], [19], [20], [15], which introduced different methods for leveraging embeddings in WSD task. For example, Chen et al. (2014) presented a unified model for word sense representation and disambiguation using WordNet sense inventory. Their model assigns distinct representations for each word sense. Taghipour and Ng (2015) proposed a semi-supervised WSD method using word embeddings to improve the WSD system. They created an adaptation of word embeddings using feed-forward neural networks and added these word embeddings to a supervised WSD system. Rothe and Schutze (2015) presented AutoExtend to learn embeddings for synsets and lexemes from standard word embeddings using WordNet as a lexical resource. Their model adds the AutoExtend feature to WSD standard features for improving performance of the system. In particular, Iacobacci and Navigli (2015) studied different techniques of combination of embeddings with standard WSD features to improve performance of conventional WSD systems. While these mentioned works use a word embeddings model of one language to disambiguate its words, we exploit a pre-trained model of one language to resolve ambiguity of another language. The detail of our method are described in next section.

## III. The proposed model

In this section, first, we discuss the overall description of the proposed unsupervised WSD system, and then the formal definition is presented in III. II.

### A. *Disambiguating scheme*

As described before, Persian words are disambiguated using trained word2vec model of the second language, i.e., English in this work. The method is based on the fact that the context provides useful information for finding the correct sense of the ambiguous word while all words in a sentence are related to each other semantically and syntactically. Therefore, having knowledge about some words, useful clues about others could be discovered.

In this study, semantic similarity is considered and is based on this hypothesis that a given polysemous word, almost all of the context words except stop words carry useful information. After preprocessing Persian raw sentences, including tokenization and stop word removal, a set $C$ is created for the ambiguous word and the context words are put into the set. Note that this set does not include the ambiguous word. Next, each word within $C$ is translated into English using a Persian-English dictionary and the set $C'$ is created. The dictionary is a bilingual word-by-word dictionary and sometimes there is more than one translation per each Persian word; in this situation, we pick up all candidate translations. Now, the context information is transferred into English. In the next step, the set $C'$ is compared against each candidate translation. An overview of the proposed model is shown in Fig. 1. As shown in Fig. 1, the Persian surrounding words $c_i$ are translated into $c'_j$, then these English-translated surrounding words are compared against candidate senses $S_i$ in the word embedding space.

### B. Formal definition

Given an ambiguous word $w$ with context words $C = \{c_1, c_2, ..., c_T\}$ excluding stop words, and possible senses of $S = \{s_1, s_2, ..., s_N\}$, in which $T$ and $N$ denote the number of content words in the context and the number of possible senses of $w$, respectively.

Let $D$ be a bilingual dictionary consisting words from the first language and their translation words in the second language. The dictionary $D$ is used to translate set $C$ to another set such as $C' = \{t_1^1, ..., t_1^{N_{c_1}}, t_2^1, ..., t_2^{N_{c_2}}, ..., t_T^1, ..., t_T^{N_{c_T}}\}$ where each $t_j^i$ represents the $i$-th candidate translation of $c_j$ and $N_{c_j}$ is the number of possible translations of word $c_j$. For simplicity assume that

$$M = \sum_{k=1}^{T} N_{c_k}, \qquad (1)$$

hence, another representation for $C'$ is:

$$C' = \{c'_1, c'_1, ..., c'_M\}. \qquad (2)$$

For vector representation of words, a function $\beta : V \mapsto R^n$ with $w \mapsto \beta(w)$ is defined; $V$ is the vocabulary and $\beta(w)$ is the vector representation of word $w$. Therefore using $\beta$ the vector representations of sets $S$ and $C'$ are $\beta(S) = \{\beta(s_1), \beta(s_2), ..., \beta(s_n)\}$ and $\beta(C') = \{\beta(c'_1), \beta(c'_2), ..., \beta(c'_M)\}$ respectively.

Similar to all similarity-based tasks, a similarity function is needed for comparing similarity between word vectors. And finally, we have $f : R^n * R^n \to R$ with $(\beta(w_i), \beta(w_j)) \mapsto f_{ij}$. To predict sense of a polysemous word, two strategies are used:

*Sum-Vec Strategy (SVS)*: In this strategy, first, the summation vector of vectors within the set $\beta(C')$ is computed. Then using a similarity measure the sum vector is compared against the vectors within the set $\beta(S)$. Assume $R$ is the sum vector of $\beta(C')$ then $F_i = f(\beta(s_i), R)$ represents similarity between $i$-th candidate translation and $R$. Thus the set $F_i = \{F_1, F_2, ..., F_N\}$ is provided and if the similarity function $f$ is cosine similarity measure, the predicted sense is determined as $s^* = argmax_{s \in S} F$.

*Each-Vec Strategy (EVS)*: For this strategy a different scheme is considered. In EVS, each vector within $\beta(S)$ is compared against each vector within $\beta(C')$ and therefore for each sense vector within $\beta(S)$ a set $F_i = \{f_{i1}, f_{i2}, ..., f_{iM}\}$ is obtained. Note that $f_{ij} = f(\beta(s_i), \beta(c'_j))$ represents similarity between $i$-th sense of $S$ and $j$-th clue word of $C'$. Alike SVS strategy, in EVS the cosine similarity measure is used. In next step, the average value of each $F_i$ is calculated, then a set $G = \{G_1, G_2, ..., G_N\}$ is obtained such that $G_i = \frac{1}{M} \sum_{j=1}^{M} f_{ij}$ and finally the sense is predicted as $s^* = argmax_{s \in S} G$.

## IV. EXPERIMENTS AND RESULTS

In this section, first we clarify some explanations about the dataset. In Section IV. II, the training procedure, the model, and used configurations are discussed. In Section IV. III, the results of experiment for two strategies of configuration 1 with different similarities are shown and finally the proposed approach is compared against MFS baseline and Kaji's method [13].

### A. Dataset

To evaluate the proposed methods, having a gold standard dataset was essential. The primary data for this study were collected from Persian Wikipedia articles containing some selected ambiguous words. Despite the difficulty of creating a new test dataset, annotating of data was done manually by authors of this work. The whole dataset and results are publicly available at https://iasbs.ac.ir/~ansari/nlp/wsdw2vec.html.

In our experiments, four Persian ambiguous words were selected. These words are "شیر" [shir], "سبک" [Sabk/Sabok], "جو" [Jo/Jav] and "جرم" [Jerm/Jorm]. For each word, we consider the two most frequent senses. Among these words "سبک", "جو" and "جرم" are heteronyms, that is, words are written alike but different in pronunciation and meaning. The word "شیر" is a homonym. Homonyms have the same pronunciation with a different meaning. The dictionary used in this study is a word-by-word bilingual Persian-English dictionary created by the authors which includes about 45K entries of Persian words such that for each entry one or more translations in English is provided.

### B. Word2vec model training

In order to train our model, we use the python implementation of word2vec embedded in Gensim [21]. Gensim is a free python library, including different modules for NLP, machine learning and data mining. When it comes to training phase there are several parameters. Among them three main parameters are manipulated, including window size, min_count, number of dimensions in mapping space, while other parameters are left with their default values. A complete list of parameters is available on the Gensim website. Thus for each parameter assignment, a configuration is created and totally, eight configurations are selected which are shown in Table I.

### C. Results

There are many similarity measures to compare word vectors. In this experiment, cosine and Euclidean measures for the Sum-Vec Strategy (SVS), the strategy where the sum vector of translated surrounding words is compared against vectors of candidate translations of the ambiguous word are evaluated. Results are shown in TABLES III and IV. For the

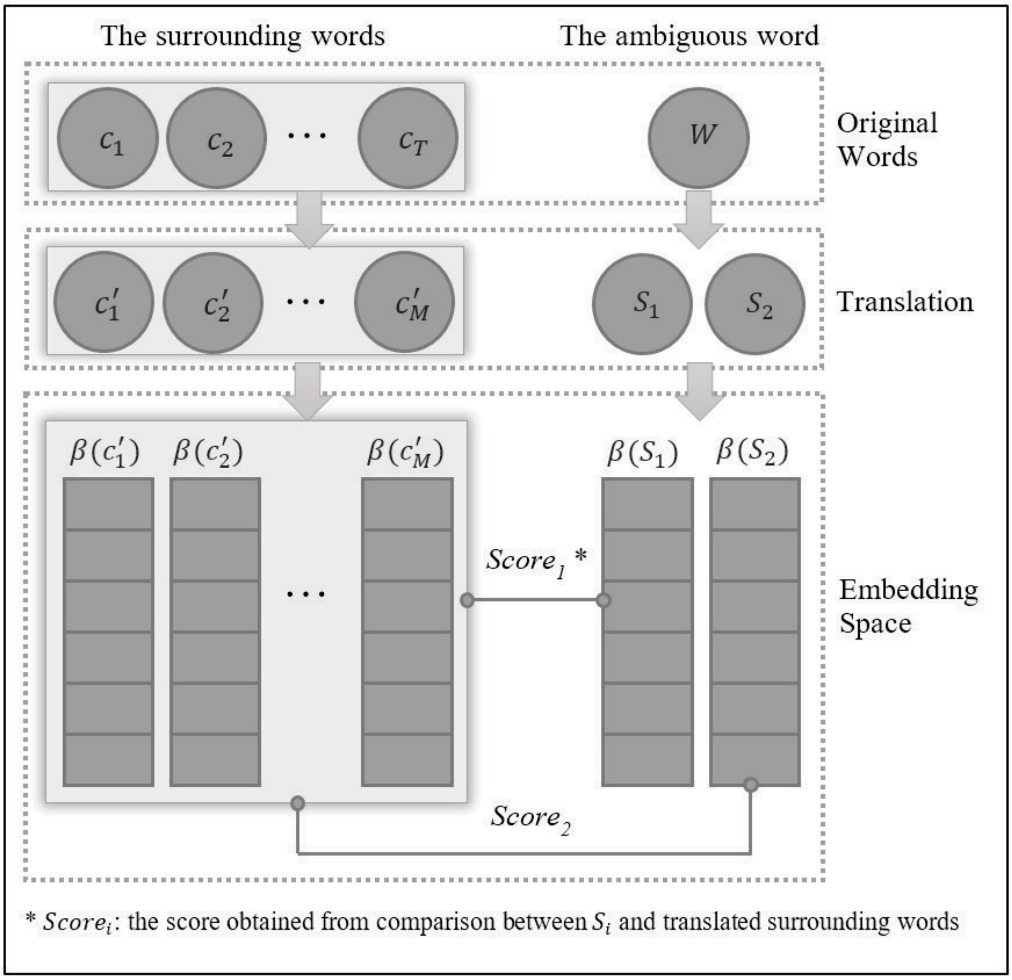


Fig. 1. An overview of the proposed model. The ambiguous word and surrounding words are translated into the second language (English in this work) and using the embedding space of the second language the correct sense is determined. There are more than one translation for the ambiguous word considering that the ambiguity will not transfer to the second language

TABLE I. LIST OF CONFIGURATIONS ARE USED IN THIS STUDY TO TRAIN OUR WORD EMBEDDING SPACE. CONFIGURATIONS 1 AND GAVE US THE BEST RESULTS IN OUR DISAMBIGUATION TASK

| Configuration | Number of Dimensions | Window Size | Min Count |
|---|---|---|---|
| 1 | 200 | 5 | 5 |
| 2 | 200 | 5 | 10 |
| 3 | 200 | 10 | 5 |
| 4 | 200 | 10 | 10 |
| 5 | 400 | 5 | 5 |
| 6 | 400 | 5 | 10 |
| 7 | 400 | 10 | 5 |
| 8 | 400 | 10 | 10 |

Each-Vec Strategy (EVS), the strategy that each translated surrounding word is compared against candidate senses, the cosine similarity is calculated and reported in Tables III and IV which represent result of Configuration 1 and Configuration 5 respectively. Configurations' details are shown in Table I). Table II shows the results of word sense disambiguation using the method presented in [13].

There are 200 samples (paragraphs or sentences) for each word. For example, "جو" has two translations in English: 'atmosphere' and 'barley'. As you see in Table V, the algorithm has predicted 189 senses correctly out of 200 samples. According to these Tables, the results represent our method as a high accuracy one. Among all of the strategies and similarity measures, Each-Vec Strategy with cosine similarity has achieved better accuracy compared to others and its overall accuracy for four words is almost 90%.

## V. CONCLUSION

The main goal of this study is to present a new unsupervised method for disambiguating Persian words using the word embeddings model of English corpora. In order to build a word embedding model, we selected English articles from the 2014 Wikipedia website. The surrounding words of the ambiguous word were translated into English using a Persian-English lexicon. Finally, each translation of the polysemous word was compared against the vectors of the English-translated surrounding words and the translation with the most similarity score was selected as the correct sense. To determine the similarity between two words, their vector representations were compared in the semantic space provided by the word2vec model.

TABLE II. Results of KAJI [13] Strategy

| Senses | شیر [Šir] | | سبک [Sabok/Sabk] | | جو [Jo/Jav] | | جرم[Jorm/Jerm] | |
|---|---|---|---|---|---|---|---|---|
| | Milk | Lion | Style | Light | Atmosphere | Barley | Mass | Crime |
| # of senses | 134 | 66 | 138 | 62 | 134 | 66 | 160 | 40 |
| # of corrects | 117 | 29 | 116 | 33 | 119 | 32 | 143 | 19 |
| | 146 | | 149 | | 151 | | 162 | |
| accuracy | 73.0 % | | 74.5 % | | 75.5 % | | 81.0 % | |

TABLE III. Results of SUM-VEC Strategy for Cosine Similarity in Configuration 1

| Senses | شیر [Šir] | | سبک [Sabok/Sabk] | | جو [Jo/Jav] | | جرم[Jorm/Jerm] | |
|---|---|---|---|---|---|---|---|---|
| | Milk | Lion | Style | Light | Atmosphere | Barley | Mass | Crime |
| # of senses | 134 | 66 | 138 | 62 | 134 | 66 | 160 | 40 |
| # of corrects | 124 | 41 | 122 | 41 | 133 | 44 | 160 | 29 |
| | 165 | | 163 | | 177 | | 189 | |
| accuracy | 82.5 % | | 81.5 % | | 88.5 % | | 94.5 % | |

TABLE IV. Results of SUM-VEC Strategy for Euclidean Similarity in Configuration 1

| Senses | شیر [Šir] | | سبک [Sabok/Sabk] | | جو [Jo/Jav] | | جرم[Jorm/Jerm] | |
|---|---|---|---|---|---|---|---|---|
| | Milk | Lion | Style | Light | Atmosphere | Barley | Mass | Crime |
| # of senses | 134 | 66 | 138 | 62 | 134 | 66 | 160 | 40 |
| # of corrects | 129 | 26 | 118 | 44 | 134 | 17 | 160 | 15 |
| | 155 | | 162 | | 151 | | 175 | |
| accuracy | 77.5 % | | 81.0 % | | 75.5 % | | 87.5 % | |

TABLE V. RESULTS OF EACH-VEC STRATEGY FOR COSINE SIMILARITY IN CONFIGURATION 1

| Senses | شیر [Šir] | | سبک [Sabok/Sabk] | | جو [Jo/Jav] | | جرم[Jorm/Jerm] | |
|---|---|---|---|---|---|---|---|---|
| | Milk | Lion | Style | Light | Atmosphere | Barley | Mass | Crime |
| # of senses | 134 | 66 | 138 | 62 | 134 | 66 | 160 | 40 |
| # of corrects | 126 | 42 | 117 | 46 | 128 | 61 | 158 | 32 |
| | 168 | | 163 | | 189 | | 190 | |
| accuracy | 84.0 % | | 81.5 % | | 94.5 % | | 95.0 % | |

TABLE VI. Results of EACH-VEC Strategy for Cosine Similarity in Configuration 5

| Senses | شیر [Šir] | | سبک [Sabok/Sabk] | | جو [Jo/Jav] | | جرم[Jorm/Jerm] | |
|---|---|---|---|---|---|---|---|---|
| | Milk | Lion | Style | Light | Atmosphere | Barley | Mass | Crime |
| # of senses | 134 | 66 | 138 | 62 | 134 | 66 | 160 | 40 |
| # of corrects | 127 | 34 | 123 | 45 | 128 | 61 | 157 | 35 |
| | 161 | | 168 | | 189 | | 192 | |
| accuracy | 80.5 % | | 84.0 % | | 94.5 % | | 96.0 % | |

Experiments on our dataset show highly accurate results and provide a significant improvement over previous unsupervised methods. Although, the efficiency of the approach was verified for word sense disambiguation of Persian texts (using English as the supplementary language) it can be applied to any other pair of languages as well.

## Acknowledgment

The authors gratefully acknowledge the contribution and help of Hamid Haghdoost, Alireza Abbas Alipour, and Mohammad Mahmoudi to this work.

The research was supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495, International Mobility of Researchers at Charles University, and by grant No. 19-14534S of the Grant Agency of the Czech Republic.

## References

[1] R. Navigli and L. Mirella, "An experimental study of graph connectivity for unsupervised word sense disambiguation." IEEE transactions on pattern analysis and machine intelligence, 2010.

[2] M. Carpuat and D. Wu, "Improving statistical machine translation using word sense disambiguation." EMNLP-CoNLL, 2007.

[3] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller, "Word-sense disambiguation for machine translation," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 771–778. [Online]. Available: https://doi.org/10.3115/1220575.1220672

[4] Y. S. Chan, H. T. Ng, and D. Chiang, "Word sense disambiguation improves statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 33–40. [Online]. Available: https://www.aclweb.org/anthology/P07-1005

[5] C. Carpineto and G. Romano, "A survey of automatic query expansion

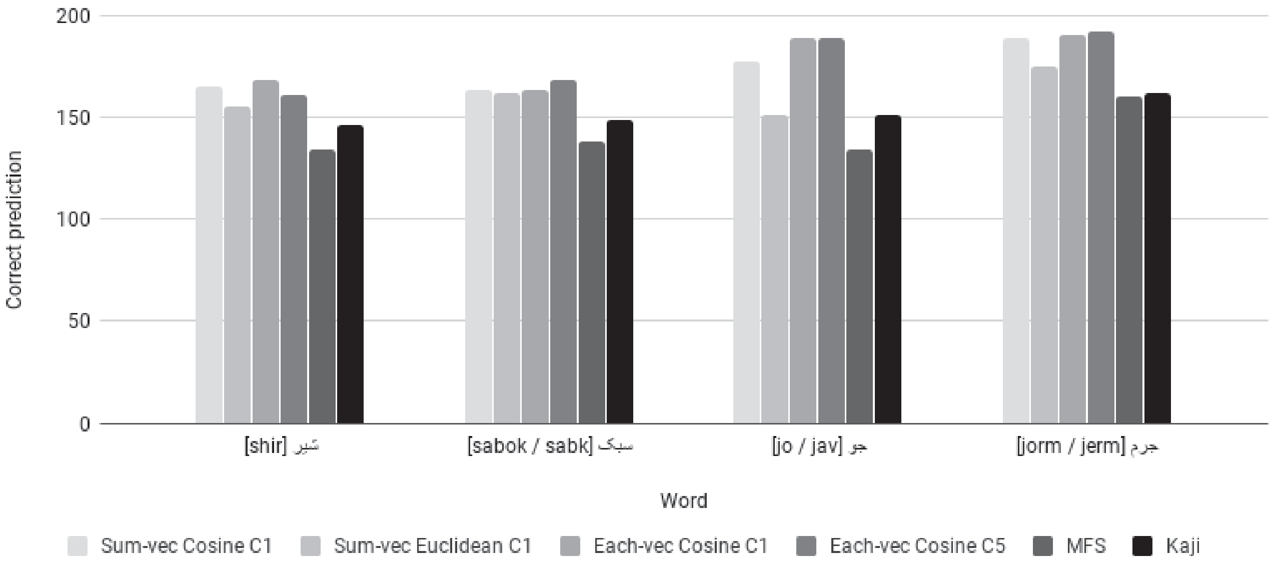


Fig. 2. Comparison of our strategies against MFS baseline and Kaji's method. Each column shows represents the number of correct predictions for 200 samples of under question word. C1 and C5 stand for configuration 1 and configuration 5, respectively

in information retrieval," *ACM Comput. Surv.*, vol. 44, no. 1, pp. 1:1–1:50, Jan. 2012. [Online]. Available: http://doi.acm.org/10.1145/2071389.2071390

[6] G. Ramakrishnan, A. Jadhav, A. Joshi, S. Chakrabarti, and P. Bhattacharyya, "Question answering via bayesian inference on lexical relations," in *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*. Sapporo, Japan: Association for Computational Linguistics, Jul. 2003, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/W03-1201

[7] L. P. De Albornoz, Jorge Carrillo and P. Gervás, "Improving statistical machine translation using word sense disambiguation." Research in Computing Science, 2010.

[8] L. Plaza, M. Stevenson, and A. Díaz, "Improving summarization of biomedical documents using word sense disambiguation," in *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, ser. BioNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 55–63. [Online]. Available: http://dl.acm.org/citation.cfm?id=1869961.1869968

[9] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation*, ser. SIGDOC '86. New York, NY, USA: ACM, 1986, pp. 24–26. [Online]. Available: http://doi.acm.org/10.1145/318723.318728

[10] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, Jun. 1995, pp. 189–196. [Online]. Available: https://www.aclweb.org/anthology/P95-1026

[11] E. Agirre and G. Rigau, "Word sense disambiguation using conceptual density," in *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, ser. COLING '96. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996, pp. 16–22. [Online]. Available: https://doi.org/10.3115/992628.992635

[12] S. Banerjee and T. Pedersen, "An adapted lesk algorithm for word sense disambiguation using wordnet." Springer, Berlin, Heidelberg, 2002.

[13] H. Kaji and Y. Morimoto, "Unsupervised word sense disambiguation using bilingual comparable corpora," in *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. [Online]. Available: https://www.aclweb.org/anthology/C02-1058

[14] X. Chen, Z. Liu, and M. Sun, "A unified model for word sense representation and disambiguation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1025–1035. [Online]. Available: https://www.aclweb.org/anthology/D14-1110

[15] M. T. P. Iacobacci, Ignacio and R. Navigli, "Sensembed: Learning sense embeddings for word and relational similarity." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015.

[16] R. Navigli, "Word sense disambiguation: A survey." ACM Computing Surveys (CSUR), 2009.

[17] A. Karimi, E. Ansari, and B. S. Bigham, "Extracting an English-Persian Parallel Corpus from Comparable Corpora," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*, 2018.

[18] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995. [Online]. Available: http://doi.acm.org/10.1145/219717.219748

[19] K. Taghipour and H. T. Ng, "Semi-supervised word sense disambiguation using word embeddings in general and specific domains," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 314–323. [Online]. Available: https://www.aclweb.org/anthology/N15-1035

[20] S. Rothe and H. Schütze, "Autoextend: Extending word embeddings to embeddings for synsets and lexemes," in *ACL*. arXiv preprint arXiv:1507.01127, 2015.

[21] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.