2020 21st International Arab Conference on Information Technology (ACIT)

Sense Inventories for Arabic Texts

Marwah Alian
Princess Sumaya University for Technology
Amman,Jordan
m.alian@psut.edu.jo

Arafat Awajan
Mutah University
Princess Sumaya University for Technology
awajan@psut.edu.jo

***Abstract*— Word sense disambiguation is the process of determining the proper meaning of a word according to its context. In this study, we represent the impact of word embedding on building Arabic sense inventory by an unsupervised approach. Three pre-trained embeddings are tested to investigate their effect on the resulting sense inventory and their efficiency in word sense disambiguation for Arabic context. Sense inventories are constructed using a fully unsupervised method based on graph-based word sense induction algorithm. The results show that Aravec-Twitter inventory achieves the best accuracy of 0.47 for 50-neighbors and a close accuracy to the Fasttext inventory for 200-neighbors.**

Keywords— Word sense induction, Word sense disambiguation, Arabic text, sense inventory.

I. INTRODUCTION

The semantic similarity has an important role in different applications of Natural Language Processing (NLP). Ambiguous words affects semantic similarity between two texts because the similarity score between texts depends on the similarity of their context words to determine if the two texts are similar or not [1] [2].

A single word that may have different meanings is called ambiguous word and the process of detecting the appropriate meaning of that word is known as Word sense disambiguation (WSD) [3]. The context of an ambiguous word consists of the words surrounding the ambiguous target word.

According to Alian et al. [4], WSD approaches can be categorized into knowledge-based, supervised, unsupervised and hybrid approaches. In the Knowledge based approaches, the different meanings of an ambiguous word are extracted from a dictionary or a lexicon. Supervised approaches used training annotated corpus and testing sets while unsupervised approach have no training set, they used word context and clustering algorithms while hybrid approaches merge between different methods.

One of the unsupervised approaches is the word sense induction approach which represents words as a graph then uses a clustering algorithm to group similar words together then each cluster is considered as a sense. Further descriptions are provided in section III.

This study utilizes one of the word sense induction approaches to build sense inventory for Arabic based on pre-trained embeddings. Then, the sense inventory is used in WSD for sentences with ambiguous words from The Arabic paraphrasing benchmark [5]. The sense inventory is then evaluated using the retrieved senses in terms of accuracy measure.

This research is organized in five sections as follows: Section II reviews the previously proposed work related to Arabic WSD. Section III explains the sense induction algorithm used for constructing sense inventory while Section IV discusses experiment and results then Section V is the conclusion.

II. RELATED WORK

Different approaches are proposed for Arabic WSD using word representation. For example; Alian et al. [6] have used Wikipedia as a lexical resource, and Vector Space Model as a representational approach to texts and then cosine similarity is used to measure the relatedness between Wikipedia's retrieved senses and the text that has an ambiguous word.

Hadni et al. [7] have utilized two external resources, Arabic WordNet (AWN) and English WordNet (WN), to translate terms that could not be found in AWN using Machine Translation System. Then, the nearest concept for the ambiguous word is chosen based on the number of relationships between concepts in the same local context. They evaluate their approach using Naïve Bayesian and Support Vector Machine (SVM). The proposed approach achieves accuracy of 0.732 using Wu and Palmer's [8] measure with SVM.

Representing words as vectors in the distributional space has attracted the researcher in different NLP applications and has provided promising results. Therefore, Arabic WSD based on word embedding is one of these applications. For example; Laatar et al. [9] have proposed a WSD method based on word embedding where the word embedding is learned using Skip-Gram model [10]. The similarity is measured between context vector and senses vectors where the context vector is computed using words embeddings that appear in the context of an ambiguous word. Senses definitions are retrieved from a dictionary and the most similar definition vector to the context vector is selected as the appropriate sense. This approach achieves accuracy of 0.78.

In addition, the work of Alkhatlana et al. [11] utilizes two embedding methods, Word2vec and GloVe, to generate global contexts of words and extracts synsets of ambiguous words from AWN. They construct a test dataset to be used for WSD task. The sense vector is obtained based on the retrieved AWN synset then the cosine similarity between context vector and sense vector is computed. The most similar sense vector is considered as the correct sense for an ambiguous word.

Logacheva et al. [12] have proposed a new unsupervised WSD approach that depends on pre-trained embeddings and does not need any external annotated corpus. In this approach, a semantic graph is constructed for words in the vocabulary of the pre-trained embeddings model then the graph is clustered into subgraphs according to the similarity between words vectors. Each subgraph represents a sense then retrofitting approach is used to make the sense vector in the direction of the ambiguous word. They have used Fasttext embeddings to build sense inventories for 158 languages including Arabic.

A comparison between the previously discussed approaches is given in Table I. The comparison includes the authors, the publication year, the category of approach, corpus or dataset used, the sense inventory, the evaluation metric and the results.

978-1-7281-8855-3/20/$31.00 ©2020 IEEE

TABLE 1: COMPARISON BETWEEN APPROACHES USED FOR WSD

Ref	Year	Approach category	Dataset/Corpus/embeddings	Sense Inventory	Similarity measure	Evaluation metric	Results
Alian et al. [6]	2016	Knowledge based	External dataset	Wikipedia	Cosine similarity	N/A	-
Hadni et al. [7]	2016	Knowledge based	Essex Arabic Summaries Corpus (EASC)	AWN, WN.	Wu and Palmer's	Accuracy	0.732
Laatar et al. [9]	2017	Semi-supervised	Historical Arabic Dictionary Corpus,	Almu-Jam-Alwasit dictionary	Cosine similarity	Precision	0.78
Alkhatlana et al. [11]	2018	Semi-supervised	collected from Arabic News	AWN	Cosine similarity	N/A	-
Logacheva et al. [12]	2020	Unsupervised	Fasttext pre-trained embeddings	Created from embeddings	Cosine similarity	N/A for Arabic	-

In this research, we apply the approach of Logacheva et al. [12] using pre-trained embeddings of Aravec. WSD is then applied to 86 sentences from the Arabic paraphrasing benchmark using the senses extracted from sense inventories. The results are compared to the results of the senses retrieved from Fasttext inventory.

III. WORD SENSE DISAMBIGUATION

The process of disambiguation of polysemy words depends on building sense inventory to retrieve the senses of the ambiguous word and then select the most similar sense to the context of an ambiguous word. We benefit from the work of Logacheva et al. [12] in building Arabic sense inventory using two different Aravec pre-trained embeddings. One is trained on Twitter and the other is trained on Wikipedia.

The algorithm of Logacheva et al. [12] consists of two main concepts; the first relies on graph-based word sense induction, while the second is the graph filtering using vector operations for word vectors.

A. Word Sense Induction

Word Sense Induction depends on finding a list of nearest neighbors for word embedding in the distributional space. The method of constructing the semantic graph is as follows:

For each word *w* in Vocabulary:

- Construct the set of N-nearest neighbors (S) for the target word w. Let S members be: $\{s_1, s_2, \ldots, s_n\}$.
- Construct the set of N-anti-neighbors (Δ) that consists of words that are not similar to the corresponding nearest neighbors of *w* where the vectors of these words is computed as the subtraction between the vector of word *w* and its neighbor *s*: (w- s_i).
- Construct the set $\overline{N} = \{\overline{s_1}, \overline{s_2}, \ldots, \overline{s_n}\}$ that consists of the most similar words to the vectors in Δ but the result may be the target word *w*
- The set of anti-pairs consists of $(s_i, \overline{s_i})$ but not the target word *w* $(\overline{s_i} \neq w)$.
- These anti-pairs are words that should not be connected in the graph unless if both words s_i and $\overline{s_i}$ are members in the set of N-nearest neighbors (S).
- Construct the set of vertices of the graph (V) by adding words from the set (S) and their anti-pair from $(\overline{N})$ only if the word and its anti-pair are part of the set (N) of the target word *w*. In other words, only add to the set (V) words that may benefit in separating different senses of *w*. [12]
- Construct the set of edges (E): for each word s_i in the nearest neighbors (S) create a set of nearest neighbors $(S') = \{c_1, c_2 \ldots, c_n\}$, and add the edge between word *si* and the nearest neighbor c_j if c_j is not an anti-pair of s_i.

There is no edge between a word s_i and its anti-neighbor in the graph because they belong to different senses.

B. Clustering

The constructed graph is clustered into subgraphs where each subgraph represents a sense of the target word. The average of the words embeddings in each subgraph represents the vector of the sense. Retrofitting is also applied to the sense vector. Each cluster represents a sense of the target word and the computed sense vector represents the keyword of that sense.

Each sense of the target word with its keyword and cluster is saved to the sense inventory.

C. Disambiguation

Sense vectors are used for WSD in Arabic text by extracting the senses of the ambiguous word from the sense inventory and then computing the context vector by averaging the vectors of context words that are most similar to the ambiguous target word. The cosine similarity is computed between the sense vector and the context vector. Then the most similar sense will be selected as the correct sense.

IV. EXPERIMENT AND RESULTS

In order to evaluate the disambiguation approach, three experiments are applied to sentences extracted from Arabic paraphrasing benchmark; the first experiment uses Aravec-twitter sense inventory while the second experiment uses Aravec-Wiki sense inventory and the final one is based on Fasttext inventory created by Logacheva et al. [12]. We have used accuracy as an evaluation metric for the correctness of the selected senses. The following subsections describe the dataset and the pre-trained embeddings that we have used.

A. Dataset

Arabic paraphrasing benchmark [5] is used in the experiment of word sense disambiguation. This benchmark is constructed based on Transformation rules for Arabic [13] [14] such as permutation, deletion, addition and others. These rules are applied to the structure of a sentence to produce a new sentence. The benchmark consists of 1010 sentence pairs

labeled for similarity and paraphrasing. However, the number of sentences containing ambiguous words is 86 sentences that we used in our experiment.

D. Aravec pre-trained embeddings

AraVec [15] is an Arabic distributed word embedding model that is trained using different resources and available online with different dimensions. The word embeddings have been obtained using the Word2Vec skip-gram and CBOW models [10].

Aravec-twitter model is trained on Arabic tweets with a vocabulary size of 145,428 and a dimension of 100 and 300 for each word vector. The documents size is 66,900,000.

Aravec-Wiki model is trained using 1,800,000 documents from World Wide Web pages with Arabic content with a vector dimension of 100 and 300 and a vocabulary size of 662,109.

E. Fasttext pre-trained embeddings

Arabic Fasttext embeddings are provided by Grave et al. [16]. These embeddings are resulted from training on Wikipedia and Common Crawl corpus. They have used an extension of the Fasttext model with subword information. This model is available online with a dimension of 300 for word vector.

F. Results and discussion

The retrieved senses are evaluated by an Arabic expert who provides each selected sense with a label as correct, incorrect. For ambiguous words that have no sense in the sense inventory, unknown label is given.

The number of target words to be disambiguated is 139 words and the accuracy is measured as the correct senses from the total senses where the unknown senses are excluded from the total number of senses as in Formula (1):

$$Accuracy = \frac{Correct\ senses}{Total\ senses - unknown\ senses} \quad (1)$$

The experiments are conducted for building the sense inventory based on words semantic graphs with N-neighbors for N=50, 100 and 200.

Table II, Table III and Table IV compare the results of each sense inventory for 50, 100, and 200 Neighbors, respectively, in terms of correct, incorrect, unknown senses and accuracy.

TABLE II RESULTS OF SENSE INVENTORIES FOR 50-NEIGHBORS

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-twitter	45	49	45	0.479
Aravec-Wiki	22	30	87	0.423
Fasttext	56	68	15	0.451

Table II shows that the accuracy of the sense inventory that is constructed based on Aravec-twitter pre-trained embeddings provides the best accuracy value of 0.48.

TABLE III RESULTS OF SENSE INVENTORIES FOR 100-NEIGHBORS

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-Twitter	36	83	20	0.303
Aravec-Wiki	38	70	31	0.352
Fasttext	42	77	20	0.353

It is shown from Table III that the accuracy of using Fasttext is better than that of Aravec-twitter inventory for 100-neighbors but it is similar to the accuracy achieved by Aravec-Wiki inventory.

TABLE IV RESULTS OF SENSE INVENTORIES FOR 200-NEIGHBORS

Sense Inventory	Correct	Incorrect	Unknown	Accuracy
Aravec-twitter	60	69	10	0.465
Aravec-Wiki	53	72	14	0.424
Fasttext	54	62	23	0.466

The results of 200-Neighbors show that the accuracy of Aravec-twitter-based inventory provides the least number of unknown senses. The Fasttext-based inventory provides a very similar accuracy value to the Aravec-twitter inventory.

V. CONCLUSION

This study provides a disambiguation approach for Arabic words using the word sense induction approach to build a sense inventory for Arabic words. Then, an evaluation for three sense inventories is provided where these inventories are based on three different pre-trained embeddings, namely, Aravec-twitter, Aravec-Wiki and Fasttext embeddings.

In the experiment of 50-neighbors sense-inventory, the Aravec-twitter sense inventory achieves the best accuracy of 0.47 whereas in the 100-neighbors experiment, the Fasttext sense-inventory provides better accuracy value.

In the case of 200-neighbors, the Aravec-twitter sense inventory and the Fasttext sense inventory achieve very similar accuracy values.

However, in the case of paraphrasing identification task, the polysemy problem still has to be studied. This requires more analysis of semantic similarity and material resources to evaluate the effect of WSD.

REFERENCES

[1] S. Srivastava and S. Govilkar, "A Survey on Paraphrase Detection Techniques for Indian Regional Languages," *International Journal of Computer Applications*, vol. 163, no. 9, pp. 0975 – 8887, 2017.

[2] M. Alian and A. Awajan, "Semantic similarity approaches- Review," in *2018 International Arab Conference on Information Technology (ACIT2018)*, Werdanye, Lebanon, 2018, pp. 1-6.

[3] R. Laatar, C. Aloulou, and L.H. Bilguith, "Word sense disambiguation of Arabic language with Word Embeddings as part of the Creation of a Historical Dictionary," in *2018 8th International Conference on Computer Science and Information Technology (CSIT)*, Amman,Jordan, 2018.

[4] M. Alian, A. Awajan, and A. Al-Kouz, "Arabic Word Sense Disambiguation-Survey," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan., 2017, pp. 236 - 240.

[5] M. Alian, A. Awajan, A. Al-Hasan, and R. Akuzhia, "Towards building Arabic paraphrasing benchmark," in *the Second International conference on Data Science, E-learning and Information Systems (DATA' 2019)*, Dubai, 2019.

[6] A. Alian, A. Awajan, and A. Al-Kouz, "Word sense disambiguation for Arabic text using Wikipedia and Vector Space Model," *International Journal of Speech Technology*, vol. 19, no. 4, pp. 857-867, 2016.

[7] M. Hadni, S. El Alaoui, and AM. Lachkar, "Word Sense Disambiguation for Arabic Text Categorization," *The International Arab Journal of Information Technology, Vol. 13, No. 1A, 2016*, vol. 13, no. 1A, pp. 215-222, 2016.

[8] Z. and Palmer, M. Wu, "Verb semantics and lexical selection," in *the 32nd Annual Meeting of the Associations for Computational Linguistics*, 1994, pp. 133–138.

[9] R. Laatar, C. Aloulou, and LH. Belguith, "Word sense disambiguation of Arabic language with Word Embeddings as part of the Creation of a Historical Dictionary," in *International Workshop on Language Processing and Knowledge Management (LPKM 2017)*, Sfax, Tunisia, 2017.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119.

[11] A. Alkhatlana, J. Kalita, and A. Alhaddad, "Word Sense Disambiguation for Arabic Exploiting Arabic WordNet and Word Embedding," in *The 4th International Conference on Arabic Computational Linguistics (ACLing 2018)*, Dubai, UAE, 2018, pp. 50-60.

[12] V. Logacheva et al., "Word Sense Disambiguation for 158 Languages using Word Embeddings Only," *arXiv:2003.06651v1* , Mar 2020.

[13] N. Chomsky, *syntactic structure*. the Hague. Paris: Mouton publishers, 1957.

[14] M. AlKouli, *Transformation Rules for Arabic Language (qwAEd tHwylyAh llgAh AlErbyAh)*. Jordan: Dar Al-Falah, 1999.

[15] A. Mohammad, K. Eissa, and S. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017.

[16] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning Word Vectors for 157 Languages," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.