

Evaluation of Stacked Embeddings for Arabic Word Sense Disambiguation

Rim Laatar, Chafik Aloulou, Lamia Hadrach Belguith

Université de Sfax,
Mir@cl laboratory,
Tunisia

rimlaatar@yahoo.fr, {chafik.aloulou, l.belguith}@fsegs.rnu.tn

Abstract. Word Sense Disambiguation (WSD) aims to determine the correct meaning of words that can have multiple interpretations. Recently, contextualized word embeddings, whose goal is to give different representations of the same word in diverse contexts, have been shown to have a tremendous impact on several natural language processing tasks including question answering, semantic analysis and even word sense disambiguation. This paper reports on experiments with different stacks of word embeddings and evaluation of their usefulness for Arabic word sense disambiguation. Word embeddings stay in the core of the development of NLP, with multiple key language models being created over the last two years like FastText, EIMo, BERT and Flair. It's worth pointing out that the Arabic language can be divided into three major historical periods: old Arabic, middle-age Arabic, and Modern Arabic. Actually, contemporary Arabic has proved to be the greatest concern of many researchers. The main gist of our work is to disambiguate Arabic words according to the historical period in which they appeared. To perform the WSD task, we suggest a method that deploys stacked embeddings models. The experimental evaluation demonstrates that stacked embeddings outperforms the previously proposed methods for Arabic WSD.

Keywords. Arabic language, historical dictionary, modern standard Arabic, old Arabic, middle age Arabic, contextualized embeddings, stacked embeddings.

1 Introduction

In natural language processing (NLP), WSD is the problem of identifying the meaning of a word by taking into account its context of use. WSD is one of the oldest challenges in the field of NLP.

Indeed, disambiguation of word senses in context is easy for humans, but is a major challenge for automatic approaches.

Arabic is a challenging language due not only to the presence of ambiguous words with diverse meanings, but also to their semantic changes and their evolution over time. In fact, multiple contributions have enriched the Arabic language throughout history.

Thus, some words disappear, while others appear. But, some changes in word meanings may lead to the appearance of more updated words that correspond to contemporary reality. Indeed, the changes of word meanings produced by the modifications imposed by social life are frequent.

For instance, the word “القهوة” refers now to the “coffee” rather than the “wine” which was its original meaning. Thus, numerous works were developed in the literature to disambiguate Arabic words ([9, 7, 6]).

All these works are concerned with identifying the meanings of modern Arabic words. However, the Arabic language can be classified into three main historical periods, namely old Arabic, middle-age Arabic and contemporary Arabic.

In light of this, it is obvious that the senses of words are not fixed. In fact, they are constantly changing and evolving due to time and events.

Therefore, it is crucial for a contemporary reader exploring texts from an earlier era to give relevant meanings to certain words depending not only on the historical situation in which these words were produced in their original texts, but also on contexts and conditions.

In this work, we propose a simple, yet effective, word sense disambiguation method that uses a combination of a lexical knowledge-base and contextualized embeddings.

In fact, we have investigated in this study the efficiency of contextual word embeddings, more precisely, stacked embeddings in order to disambiguate Arabic words.

Our method aims to train a neural embedding model based on the Flair architecture [2] to be able to identify the meaning of an ambiguous word with reference to the context of its occurrence in a particular text. Therefore, the major contributions of this paper are as follows:

- Pre-train the FLAIR model specifically for the Arabic language based on the Historical Arabic Dictionary Corpus (HADC) [5],
- Put forward a method, which helps to automatically extract the meaning of a given term that occurred in a particular context.

The main purpose of this method is not only to disambiguate the given word according to its context of use but also according to the era in which it occurred,

- Study different strategies in order to build context vectors and sense vectors,
- Show that the stacked embedding [2] technique achieves the best disambiguation performance.

The remainder of the paper is structured as follows: in section 2, we present a survey on the main approaches to word sense disambiguation that have been widely proposed.

We also give an overview of the related works, which focused on Arabic word sense disambiguation based on contextualized embeddings.

In Section 3, we illustrate our proposed method for Arabic word sense disambiguation. Section 4 presents our experiments and results. We finally draw a conclusion and future work directions in Section 5.

2 Related Works

Two different categories of machine learning and knowledge-based methods have been explored over the years to tackle the WSD problem [27]. On the one hand, knowledge based methods use a myriad of lexical-semantic resources, such as dictionaries, ontologies or thesaurus.

An example of such a method is the Lesk algorithm [19] that aims to disambiguate a word by choosing the sense with the greatest number of overlapping words between the definitions of each of the meanings of the target word, and those of the context words.

On the other hand, however, machine learning based methods exploit annotated and unannotated corpora to disambiguate words. These methods can be divided into supervised, semi-supervised and unsupervised learning approaches.

Supervised methods use large quantities of examples from sense-annotated corpora. The trained model is then used to assign the correct sense to each word in a given context.

Yet, unsupervised methods exploit a large amount of unlabeled data rather than manually sense-tagged corpora in order to disambiguate a word in a particular context. Semi-supervised methods use some annotated data to create a large semantically annotated corpus.

Among various approaches to the WSD task used over the past two decades, a supervised learning approach has been the most successful.

However, it is quite expensive in both time and cost to annotate a large amount of data because supervised WSD requires a large amount of manually labeled training examples to achieve good performance.

Unsupervised learning approaches, however, require neither labeled examples nor diverse resources. Thus, these approaches are typically less accurate than supervised algorithms because examples may not be assigned the correct sense.

During recent years, the use of contextualized embeddings that produce different representations for the same word depending on its contextual usage [11, 24, 3], has contributed to a series of significant advances in a range of NLP tasks, such as Named Entity Recognition [29], Sentiment

Table 1. Main characteristics of the HADC corpus

Historical period	Number of text
Pre-Islamic era	100
Islamic era	101
Abbasid era	383
Middle era	147
Modern era	138
Total	869

Analysis [21] and WSD [26]. In fact, contextualized word embeddings are reported to be highly powerful as they represent words as vectors varying across linguistic contexts. This permits them to capture more complex characteristics of word meaning, including polysemy [14].

The established English success makes contextualized embeddings an attractive option for Arabic consideration. In this concern, many contextual embedding models have been developed, like BERT [11], RoBERTa [20], ALBERT [18], hULMonA [13] and AraBERT [8].

Accordingly, many works have investigated contextualized embeddings to solve WSD problems. However, works focusing on Arabic word sense disambiguation based on contextualized embeddings are relatively limited compared with other languages.

Some studies have recently implemented different embedding models for Arabic word sense disambiguation. [4] evaluated the performance of using word2vec and Lemma2Vec models for modern Arabic word disambiguation.

They constructed different models based on two different corpora, and they tuned different types of parameters. The final results show that Lemma2Vec models are slightly better than Word2Vec models for Arabic word disambiguation. [12] presented an Arabic gloss-based WSD technique.

They utilized the Bidirectional Encoder Representation from Transformers (BERT) to build two different models that can efficiently perform Arabic WSD. The authors used the models proposed in [8, 1] to build two gloss-based WSD models. The first model uses the pre-trained BERT models as a feature extractor without

fine-tuning BERT layers to generate a contextual word embedding of the target word in its context. They also used it to generate a sentence vector representation of the definition sentence.

These representations' vectors were then fed to a trainable dense layer to perform supervised WSD. In the second model, they fine-tuned BERT layers by training them with a sentence pair classification objective.

[16] trained a neural language model for Arabic language based on the Flair embeddings technique [3]. The main idea of their work is to study the role of different training parameters of the neural network in WSD performance.

Then, they developed a method that helps to automatically extract the meaning of a given Arabic term. Their proposed method consists in building a recurrent neural network model in order to calculate a distributed representation of both the contexts of use of the ambiguous term and its corresponding definitions.

It should be noted that all the previous methods that focused on Arabic Word Sense Disambiguation were just concerned with identifying the meaning of terms in modern Arabic. Only [16] focused on disambiguation of Arabic terms according to the distinct historical period in which they occurred.

However, they were limited to evaluate the Flair technique during the disambiguation process. Hence, the idea of disambiguating old and middle-age Arabic items by combining multiple embeddings is by no means original.

3 Arabic WSD Method Using Stacked Embeddings

This study is conducted based on two major motivating factors. First, the success of contextualized embeddings in English encourages us to perform a similar work in Arabic for further validation. Second, the power of contextualized embeddings, especially stacked embeddings, should be used to solve the Arabic WSD problem and disambiguate words according to both their contextual appearance in a source text and the era in which they emerged.

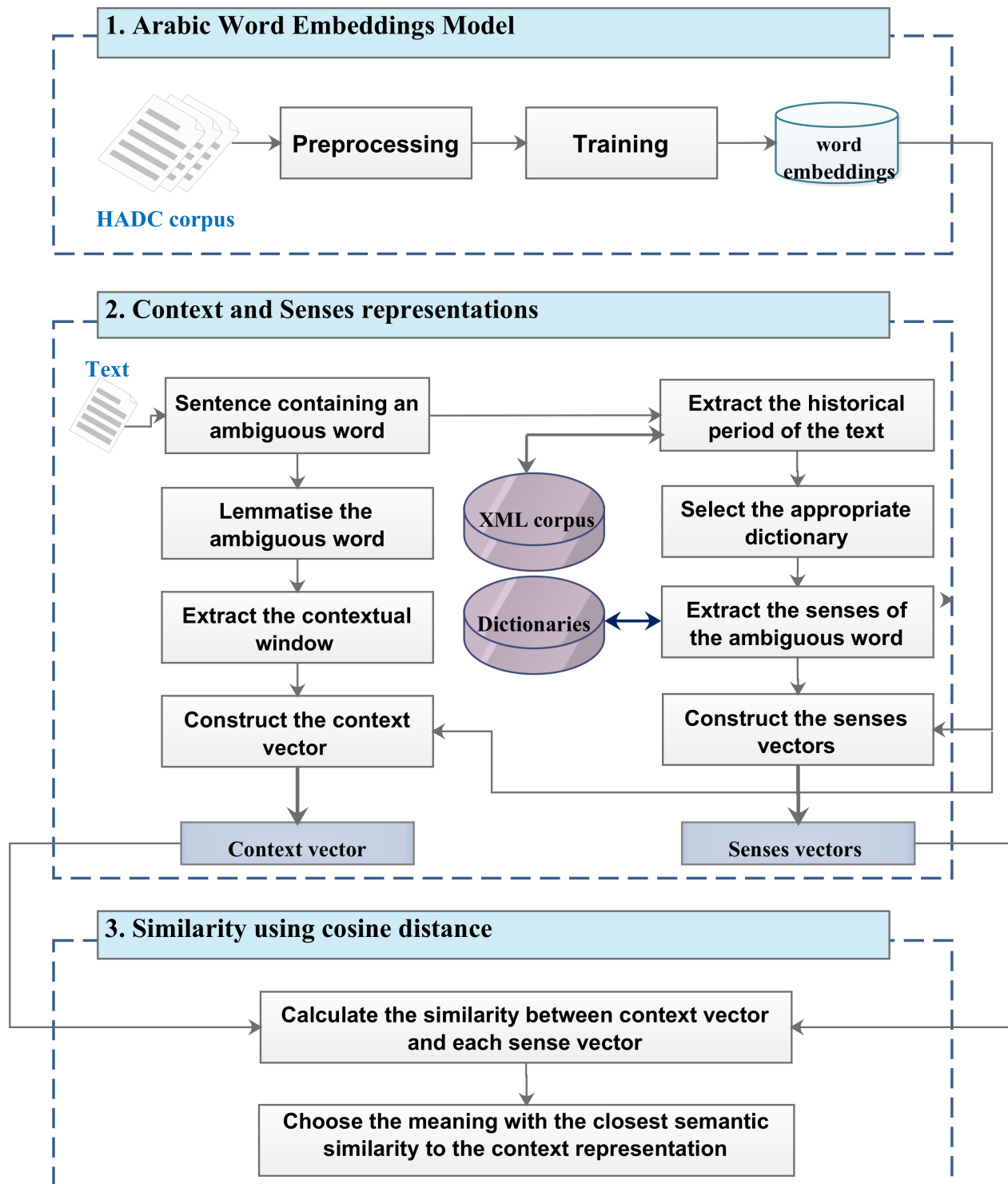


Fig. 1. The different steps of the proposed method

In this section, we describe the details of the proposed WSD method using stacked embeddings. The rationale behind this method is to embed the contextual uses and meanings of an ambiguous word taken from Arabic dictionaries.

Alongside, we applied the cosine similarity distance to determine which of the possible word senses for the target item is closer to the context representation.

The strong point of this method is that it only requires an external knowledge source and large unlabeled corpus and does not rely on labeled training data.

The aforementioned steps of the proposed method are illustrated in Figure 1. We describe each step of our method in the next sections.

3.1 Arabic Word Embedding Model

In this step, our primary objective is to train our own contextualized word embeddings based on the Flair framework [2]. In fact, Flair is an NLP library implemented by Zalando Research¹.

It is built on top of PyTorch² that provides access to many other state-of-the-art language models, such as FastText [10], GloVe [23], Elmo [24], and BERT [11].

Moreover, this technique essentially aims to learn character-level representations of a word in order to obtain dynamic vector representations for this word depending on its contexts of use.

To build a Flair Embedding model for Arabic, we have chosen the Historical Arabic Dictionary Corpus (HADC) [5]. It is originally designed to create a historical Arabic dictionary.

Indeed, this corpus contains texts at various periods, and in different domains and geographical distributions. The main characteristics of the HADC corpus are shown in Table 1.

Before training the Flair model, several normalization and preprocessing steps were performed. Infact, all diacritics, punctuations, Madda character, digits (Hindi and Arabic), Latin characters (including accented letters) were removed. Thereafter, we are going to use an LSTM with 1024 hidden states and one layer.

¹<https://github.com/flairNLP/flair>

²<https://pytorch.org/>

3.2 Context Representation

The goal behind this is to represent the context of the ambiguous word with the help of a vector. To attain this purpose, we employed two different methods, namely Document Pool Embeddings and Stacked Embeddings [2].

The first method aims to do a pooling operation over all word embeddings in a sentence to obtain an embedding for the whole sentence. In our case, the sentence represents the context in which the ambiguous word is used.

The second method aims to embed the ambiguous word in its context of use based on stacked embeddings. In fact, stacked embeddings are one of the most important concepts of the Flair library that aims at concatenating together language models in order to achieve better results.

Indeed, according to [2] stacking the embeddings can provide a powerful embedding to represent words. In this respect, we have stacked different embeddings in order to create a context vector of the ambiguous word. In the first place, we have combined Flair embedding with GloVe [23] and in the second place Flair with word2vec [22].

3.3 Sense Representation

To extract the senses of an ambiguous word, we have relied on well-defined resources, namely four Arabic dictionaries that describe the different historical periods of the Arabic language.

- For Old Arabic Dictionaries: we adopted Tahdhib Allougha Dictionary³ by Abou Mansour Azhari.
- For Intermediate Arabic Dictionaries: we used Tej Alarous Dictionary⁴ by Murtadha Zbidi.
- Concerning Modern and contemporary Arabic Dictionaries: we employed Contemporary Arabic Language Dictionary⁵.

³AlAzhari, Abu Mansour, Refining the Language. Dar Alamaarif, Cairo, 1976.

⁴Zabidi, Sayed Mortadha, Tej-Alarous, Kuwait Government Press and the National Council for Culture and Arts, Kuwait from 1965 to 2002.

⁵Mokhtar, Omar Ahmed, Modern Arabic Language, The Universe of Books, Cairo, 2008.

Table 2. Extract from the Contemporary Arabic Language dictionary before and after before and after the structuring stage

TXT Dictionary	XML Dictionary
<p>شَبَكَةٌ = شَبَكَةٌ ، جمع شَبَكَاتٌ وشَبَكَةٌ: هَدِيَّةٌ يُقَدَّمُهَا الْخَطِيبُ إِلَى خَطِيئَتِهِ إِعْلَانًا لِلخَطْبَةِ .</p> <p>شَبَّكَ = شَبَّكَ :- صَيَّادٌ بِالشَّبَاكِ . 2 - صَانِعٌ الشَّبَاكِ أَوْ بَائِعُهَا .</p>	<pre> <word id="شَبَّكَ"> <sense> <ids>1</ids> <def>شَبَّكَ صَيَّادٌ بِالشَّبَاكِ </def> </sense> <sense> <ids>2</ids> <def>صَانِعٌ الشَّبَاكِ أَوْ بَائِعُهَا </def> </sense> </word> <word id="شَبَكَةٌ"> </pre>

So, one of the most important parts of our method consists in building the intended lexical dictionary. In fact, for old Arabic, we have used Tahdhib Alougha³ Dictionary.

Moreover, we have semi-automatically developed a structured electronic dictionary with an XML format containing the glosses of 100 ambiguous old Arabic words.

Likewise, we have developed a dictionary that contains the glosses of 100 ambiguous words extracted from Tej-Alarouss⁴.

Still, the last two dictionaries, Tahdhib Alougha and Tej-Alarous, are manually structured because they have complex structures which vary from one entry to another and they are characterized by a quasi-absence of markers.

For words in modern Arabic, we have relied on Contemporary Arabic Language⁵ dictionary. Indeed, we have an HTML version of this dictionary. The latter is distinguished by a set of markers facilitating the transformation of its raw content to a structured version in XML.

Then, we automatically converted it to a structured electronic XML format. For example, Table 2 describes the structures of Contemporary Arabic Language dictionary before and after the structuring step.

Thus, our primary objective is to identify the different meanings of an ambiguous word based on the appropriate dictionary by taking into account the historical period in which this term appeared in the document.

Moreover, it is essential to know the historical period during which this word was used before constructing the appropriate vector for each definition of the ambiguous word.

In order to carry out this sub-step, we focused on the title of the document which takes the following form: The date of death followed by the name of the work.

The latter will be used, in this sense, to determine the historical period during which the document, containing the word to be disambiguated, was created.

It is important to note that the date of the author's death is of considerable importance in our work as it can embody the most precise history of the author's productions. In fact, we have not relied on the author's date of birth due to two major reasons.

First, this date is sometimes unknown especially for old authors. Second, the date of birth does not reflect the author's notoriety that most often begins after his death leaving his imprints behind him as a whole archive to be consulted.

Therefore, the historical period in which the document was produced is reflected in the most logical and objective way through the date of death.

Then, after obtaining all the sense definitions of the word to be disambiguated in the last step, we will represent each sense as a vector by using Document pool embedding and Stacked embedding methods previously introduced.

Table 3. The average precision obtained with Flair contextual embeddings and Stacked embeddings techniques

Method	Old Arabic	Middle age Arabic	Modern Arabic
Document Pool Embedding (Flair)	43.40%	42.54%	46.34%
Stack embeddings (Flair + GloVe)	49.53%	50.43%	66,18%
Stack embeddings (Flair + word2vec)	52.2%	55.34%	70.86%

3.4 Similarity Using Cosine Distance

To attribute for each ambiguous word its appropriate sense, we chose the meaning with the closest semantic similarity to the context representation. The context vectors can then be compared to the possible word sense vectors for the abstruse item.

To measure the similarity between context vectors and sense vectors, we used a cosine distance metric. In fact, the similarity measure between two vectors, $V = (v_1, v_2, \dots, v_n)$ and $W = (w_1, w_2, \dots, w_n)$, can be calculated by the cosine distance metric that is defined as follows [28]:

$$\cos(V, W) = \frac{\sum_{i=1}^n v_i \cdot w_i}{\sqrt{\sum_{i=1}^n v_i^2} \sqrt{\sum_{i=1}^n w_i^2}}. \quad (1)$$

4 Evaluation and Discussion

4.1 Code and Data

Our test corpus comprises 183 texts belonging to different historical periods. These texts have been extracted from the Historical Arab Dictionary Corpus (HADC) [5] and the Open Source Arabic Corpora (OSAC) [25].

Indeed, the Historical Arab Dictionary Corpus is divided into two main parts: one for learning and the other for testing. About 149 texts in old and medieval Arabic were specified and used for the test.

As for Modern Arabic, along with the texts extracted from the HADC, we have extracted some texts from the OSAC[25]. Moreover, we have trained the Flair embeddings model and executed the designed algorithms using the "Google Colaboratory"⁶ platform.

⁶https://colab.research.google.com/?utm_source=scs-index

The hyperparameters used to train the model are as follows:

- sequence_length = 10,
- mini_batch_size = 16 ,
- max_epochs = 20.

Indeed, this configuration allows us to obtain, after several experiments, the best embedding model that leads to better performance on the Arabic word disambiguation task in different historical periods.

Moreover, we have used Word2vec⁷ toolkit to learn vectors and Gensim⁸ to implement the model. As for GloVe we have taken the model proposed by Zalando Research⁹.

4.2 Results and Discussion

In order to measure the performance of our method, we have used the precision metric. In our case, for any ambiguous word, this metric measures the number of contexts correctly divided by the total number of annotated contexts.

During the evaluation process, we relatively consider the historical period in which an ambiguous term emerged. The major purpose of our experiment consists in evaluating different stacks of word embeddings and testing their usefulness for disambiguating Arabic terms with reference to the historical period in which they turned out.

The results of our first experiment, in which Flair contextual embeddings and pooling techniques were used, are presented in the Table 3. Then, we studied the effect of stacking different embeddings. Table 3 shows the results of combining Flair embeddings with GloVe and word2vec.

⁷code.google.com/archive/p/word2vec/

⁸<https://pypi.org/project/gensim/>

⁹<https://github.com/flairNLP/flair>

Therefore, combining word2vec [22] with Flair embeddings seems particularly beneficial for each historical period of the Arabic Language.

It should be noted, however, that combining GloVe [23] and Flair embeddings is better than the FLAIR embeddings alone.

Indeed, Table 3 illustrates that the stacked embedding method gives a better representation of the context containing the ambiguous word, and therefore performs better disambiguation results.

Consequently, using contextual embeddings, more particularly stacked embeddings, improved the results of disambiguating Arabic words according to each historical period during which these terms were used.

In summary, based on the aforementioned results, stacked embeddings represent an effective way to solve the Arabic word sense disambiguation problem. Moreover, the best results were obtained using stacked embeddings, more precisely by combining Flair with word2vec.

This is because the latter allow a better representation of the corpus terms using vectors, and subsequently a better representation of the context containing the ambiguous word with its different definitions. Therefore, improving context vectors and definition vectors leads to better results yielded by our proposed method.

Finally, we have compared the results obtained by the proposed method with the method developed by [16] and the approach proposed by [12] (Table 4).

The attained precision, which noticeably outperforms [16], is still less than the method proposed by [12]. We think that this difference might be due to the nature of ambiguous words and its complex contexts of use taken for the test. Thus, this contexts that appeared in a modern documents may have old meanings.

Recalling that we have based our comparison on the same test data, our method has given better results compared to that proposed by [16] and [17]. This result can confirm the good choice and performance of using stacked word embeddings in the Arabic Word Sense Disambiguation field.

Table 4. Comparison with others methods

Method	Result
Our Work	70.86%
[16]	66%
[12]	89%
[17]	56.45%
[15]	59.42%

5 Conclusion

This work evaluated the use of contextualized embeddings for disambiguating Arabic words. The main purpose of this study is to extract the meaning of a given word that appeared in the document.

More importantly, our method focuses equally on disambiguating not only words in Modern Arabic, but also words that emerged in ancient and middle-age Arabic periods.

We tested several embeddings including stacked embeddings in order to identify the perfect combination to achieve the best results.

The experiments show that combining word2vec and Flair word embeddings reaches a precision of 70.86% for Modern Arabic, 52.2% for Old Arabic and 55.34% for Middle-age Arabic.

During our experimentation, we have noticed that some words have meanings that existed in the corpus rather than in the dictionary. As a future work, we will try to overcome this problem by proposing an unsupervised Arabic Word Sense Disambiguation based on contextualized embeddings.

Furthermore, we will intend to look into the induction of the different senses of a given Arabic word based on the HADC corpus.

References

1. **Abdul-Mageed, M., Elmadany, A., Nagoudi, E. M. B. (2021).** ARBERT and marbert: Deep bidirectional transformers for arabic. DOI: 10.48550/ARXIV.2101.01785.

2. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 54–59. DOI: 10.18653/v1/N19-4010.
3. Akbik, A., Blythe, D., Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. Proceedings of the 27th International Conference on Computational Linguistics, pp. 1638–1649.
4. Al-Hajj, M., Jarrar, M. (2021). LU-BZU at SemEval-2021 task 2: Word2vec and lemma2vec performance in arabic word-in-context disambiguation. Proceedings of the 15th International Workshop on Semantic Evaluation, pp. 748–755. DOI: 10.18653/v1/2021.semeval-1.99.
5. Al-Said, A. B., Medea-García, L. (2014). The historical arabic dictionary corpus and its suitability for a grammaticalization approach. 5th International Conference in Linguistics.
6. Alian, M., Awajan, A., Al-Kouz, A. (2016). Arabic word sense disambiguation using wikipedia. International Journal of Computing and Information Sciences, Vol. 12, No. 1, pp. 61–66. DOI: 10.21700/ijcis.2016.108.
7. Alkhatlan, A., Kalita, J., Alhaddad, A. (2018). Word sense disambiguation for arabic exploiting arabic WordNet and word embedding. The 4th International Conference on Arabic Computational Linguistics, Vol. 142, pp. 50–60. DOI: 10.1016/j.pocs.2018.10.460.
8. Antoun, W., Baly, F., Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pp. 9–15.
9. Batita, M. A., Zrigui, M. (2019). The extended Arabic WordNet: A case study and an evaluation using a word sense disambiguation system. Proceedings of the 10th Global Wordnet Conference, pp. 46–53.
10. Bojanowski, P., Grave, E., Joulin, A., and Tomas Mikolov (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistic (TACL).
11. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. North American Chapter of the Association for Computational Linguistics (NAACL). DOI: 10.48550/ARXIV.1810.04805.
12. El-Razzaz, M., Fakhr, M. W., Maghraby, F. A. (2021). Arabic gloss WSD using BERT. Applied Sciences, Vol. 11, No. 6, pp. 2567. DOI: 10.3390/app11062567.
13. ElJundi, O., Antoun, W., Droubi, N. E., Hajj, H., El-Hajj, W., Shaban, K. (2019). hULMonA: The universal language model in arabic. Proceedings of the Fourth Arabic Natural Language Processing Workshop, Association for Computational Linguistics. DOI: 10.18653/v1/w19-4608.
14. Hofmann, V., Pierrehumbert, J., Schütze, H. (2021). Dynamic contextualized word embeddings. DOI: 10.18653/v1/2021.acl-long.542.
15. Laatar, R., Aloulou, C., Belghuith, L. H. (2018). Word embedding for arabic word sense disambiguation to create a historical dictionary for arabic language. 8th International Conference on Computer Science and Information Technology. DOI: 10.1109/csit.2018.8486159.
16. Laatar, R., Aloulou, C., Belghuith, L. H. (2020). Disambiguating arabic words according to their historical appearance in the document based on recurrent neural networks. ACM Transactions on Asian and Low-Resource Language Information Processing, Vol. 19, No. 6, pp. 1–16. DOI: 10.1145/3410569.
17. Laatar, R., Aloulou, C., Belghuith, L. H. (2020). Towards a historical dictionary for Arabic language. International Journal of Speech Technology, Vol. 25, No. 1, pp. 29–41. DOI: 10.1007/s10772-020-09704-z.
18. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. DOI: 10.48550/ARXIV.1909.11942.
19. Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries. Proceedings of the 5th Annual International Conference on Systems Documentation. DOI: 10.1145/318723.318728.

20. **Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019).** RoBERTa: A robustly optimized BERT pretraining approach. DOI: 10.48550/ARXIV.1907.11692.
21. **Mekki, A. E., Mahdaouy, A. E., Berrada, I., Khoumsi, A. (2021).** Domain adaptation for arabic cross-domain and cross-dialect sentiment analysis from contextualized word embedding. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2824–2837. DOI: 10.18653/v1/2021.naacl-main.226.
22. **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013).** Efficient estimation of word representations in vector space. Proceedings of the International Conference on Learning Representations. DOI: 10.48550/ARXIV.1301.3781.
23. **Pennington, J., Socher, R., Manning, C. (2014).** Glove: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1532–1543.
24. **Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018).** Deep contextualized word representations. North American Chapter of the Association for Computational Linguistics.
25. **Saad, M., Ashour, W. (2010).** OSAC: Open source arabic corpora. International Conference on Electrical and Computer Systems.
26. **Saeidi, M., Milios, E., Zeh, N. (2021).** Contextualized knowledge base sense embeddings in word sense disambiguation. International Conference on Document Analysis and Recognition, pp. 174–186. DOI: 10.1007/978-3-030-86159-9_12.
27. **Scarlina, B., Pasini, T., Navigli, R. (2020).** SensEmbBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. Vol. 34, No. 5, pp. 8758–8765. DOI: 10.1609/aaai.v34i05.6402.
28. **Shashavali, D., Vishwjeet, V., Kumar, R., Mathur, G., Nihal, N., Mukherjee, S., Patil, S. V. (2019).** Sentence similarity techniques for short vs variable length text using word embeddings. *Computación y Sistemas*, Vol. 23, No. 3, pp. 999–1004. DOI: 10.13053/cys-23-3-3273.
29. **Zhou, Y., Ju, C., Caufield, J. H., Shih, K., Chen, C. Y. C., Sun, Y., Chang, K. W., Ping, P., Wang, W. (2021).** Clinical named entity recognition using contextualized token representations. Computing Research Repository. DOI: 10.48550/arXiv.2106.12608.

*Article received on 08/06/2022; accepted on 08/02/2023.
Corresponding author is Rim Laatar.*