LU-BZU at SemEval-2021 Task 2: Word2Vec and Lemma2Vec performance in Arabic Word-in-Context disambiguation

Moustafa Al-Hajj
Lebanese University
Lebanon
moustafa.alhajj@ul.edu.lb

Mustafa Jarrar
Birzeit University
Palestine
mjarrar@birzeit.edu

Abstract

This paper presents a set of experiments to evaluate and compare between the performance of using CBOW Word2Vec and Lemma2Vec models for Arabic Word-in-Context (WiC) disambiguation without using sense inventories or sense embeddings. As part of the SemEval-2021 Shared Task 2 on WiC disambiguation, we used the `dev.ar-ar` dataset (2k sentence pairs) to decide whether two words in a given sentence pair carry the same meaning. We used two Word2Vec models: Wiki-CBOW, a pre-trained model on Arabic Wikipedia, and another model we trained on large Arabic corpora of about 3 billion tokens. Two Lemma2Vec models was also constructed based on the two Word2Vec models. Each of the four models was then used in the WiC disambiguation task, and then evaluated on the SemEval-2021 `test.ar-ar` dataset. At the end, we reported the performance of different models and compared between using lemma-based and word-based models.

1 Introduction

As a word may denote multiple meanings (*i.e.*, senses) in different contexts, disambiguating them is important for many NLP applications, such as information retrieval, machine translation, summarization, among others. For example, the word "table" in sentences like "I am cleaning the table", "I am serving the table", "I am emailing the table", refer to "furniture", "people", and "data" respectively. Disambiguating the sense that a word denotes in a given sentence is important for understanding the semantics of this sentence.

To automatically disambiguate word senses in a given context, many approaches have been proposed based on supervised, semi-supervised, or unsupervised learning models. Supervised and semi-supervised methods rely on full, or partial, labeling of the word senses in the training corpus to construct a model (Lee and Ng, 2002; Klein et al., 2002). On the other hand, unsupervised approaches induce senses from unannotated raw corpora and do not use lexical resources. The problem in such approaches, is that unsupervised learning of word embeddings produces a single vector for each word in all contexts, and thus ignoring its polysemy. Such approaches are called static Word Embeddings. To overcome the problem, two types of approaches are suggested (Pilehvar and Camacho-Collados, 2018): multi-prototype embeddings, and contextualized word embeddings. The latter suggests to model context embeddings as a dynamic contextualized word representation in order to represent complex characteristics of word use. Proposed architectures such as ELMo (Peters et al., 2018), ULMFiT (Howard and Ruder, 2018), GPT (Radford et al., 2018), T5 (Raffel et al., 2019), and BERT (Devlin et al., 2018), achieved breakthrough performance on a wide range of natural language processing tasks. In multi-prototype embeddings, a set of embedding vectors are computed for each word, representing its senses. In (Pelevina et al., 2017), multi-prototype embeddings are produced based on the embeddings of a word. As such, a graph of similar words is constructed, then similar words are grouped into multiple clusters, each cluster representing a sense. As for Mancini et al. (2016), multi-prototype embeddings are produced by learning word and sense embeddings jointly from both, a corpus and a semantic network. In this paper we aim at using static word embeddings for WiC disambiguation.

Works on Arabic Word Sense Disambiguation (WSD) are limited, and the proposed approaches are lacking a decent or common evaluation framework. Additionally, there are some specificities of the Arabic language that might not be known in other languages. Although polysemy and disambiguating are challenging issues in all languages,

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 748–755
Bangkok, Thailand (online), August 5–6, 2021.

they might be more challenging in the case of Arabic (Jarrar et al., 2018; Jarrar, 2021) and this for many reasons. For example, the word šāhd (شاهد) could be šāhid (شاهِد) which means a *witness*, or šāhada (شاهَدَ) which means *watch*. As such, disambiguating words senses in Arabic, is similar to disambiguating senses of English words written without vowels. Second, Arabic is a highly inflected and derivational language. As such, thousands of different word forms could be inflected and derived from the same stem. Therefore, words in word embeddings models will be considered as different, which may affect the accuracy and the utility of their representation vectors, as the same meaning could be incarnated in distributed word forms in corpora, which has led some researchers to think that using lemma-based models might be better than using word-based embeddings in Arabic (Salama et al., 2018; Shapiro and Duh, 2018). This idea will be discussed later in sections 5 and 6.

Alkhatlan et al. (2018) suggested an Arabic WSD approach based on Stem2Vec and Sense2Vec. The Stem2Vec is produced by training word embeddings after stemming a corpus, whereas the Sense2Vec is produced based on the Arabic WordNet sense inventory, such that each synset is represented by a vector. To determine the sense of a given word in a sentence, the sentence vector is compared with every sense vector, then the sense with maximum similarity is selected.

Laatar et al. (2017) did not use either stemming or lemmatization. Instead, they proposed to determine the sense of a word in context by comparing the context vector with a set of sense vectors, then the vector with the maximum similarity is selected. The context vector is computed as the sum of vectors of all words in a given context, which are learnt from a corpus of historical Arabic. On the other hand, sense vectors are produced based on dictionary glosses. Each sense vector is computed as the sum of vectors (learnt from the historical Arabic corpus) of all words in the gloss of a word.

Other approaches to Arabic WSD (Elayeb, 2019) employ other techniques in machine learning and knowledge-based methods (Bouhriz et al., 2016; Bousmaha et al., 2013; Soudani et al., 2014; Merhbene et al., 2014; Al-Maghasbeh and Bin Hamzah, 2015; Bounhas et al., 2015).

In this paper, we present a set of experiments to evaluate the performance of using Lemma2Vec vs CBOW Word2Vec in Arabic WiC disambiguation. The paper is structured as follows: Section 2 presetns the background of this work. Section 3 overviews the WiC disambiguation system. Section 4 and Section 5, respectively, present the Word2Vec and Lemma2Vec models. In Section 6 we present the experiments and the results; and in section 7 we summarize our conclusions and future work.

2 Background

Experiments presented in this paper are part of the SemEval shared task for Word-in-Context disambiguation (Martelli et al., 2021).

The task aims at capturing the polysemous nature of words without relying on a fixed sense inventory. A common evaluation dataset is provided to participants in five languages, including Arabic, our target language in this paper. The dataset was carefully designed to include all parts of speeches and to cover many domains and genres. The Arabic dataset (called multilingual `ar-ar`) consists of two sets: a train set of 1000 sentence pairs for which tags (TRUE or FALSE) are revealed, and a test set of 1000 sentence pairs for which tags were kept hidden during the competition. Figure 1 gives two examples of sentence pairs in the `dev.ar-ar` dataset. Each sentence pair has a word in common for which start and end positions in sentences are provided. Participants in the shared task were asked to infer whether the target word carries the same meaning (TRUE) or not (FALSE) in the two sentences.

{"id": "dev.ar-ar.0",
"lemma": "مَلاك",
"pos": "NOUN",
"sentence1": " ونظرا لأهمية هذه المسائل لسير عمل المحكمة مستقبلا، يلزم توفير ملاك كاف من الموظفين منذ بدء عملياتها.",
"sentence2": " ولا توجد حراسة أمام جميع البعثات الدبلوماسية بسبب النقص المزمن في ملاك أفراد الشرطة في مقر قيادتها في العاصمة."}
{"id": "dev.ar-ar.1",
"lemma": "مَلاك",
"pos": "NOUN",
"sentence1": " ونظرا لأهمية هذه المسائل لسير عمل المحكمة مستقبلا، يلزم توفير ملاك كاف من الموظفين منذ بدء عملياتها.",
"sentence2": " وأعربت عن رغبتها في الحصول على معلومات بشأن موارد هاتين الوزارتين وملاكهما وبشأن لجان إعادة التأهيل وإعادة التوطين والمساعدة الوارد ذكرها في العرض الشفوي."}

Figure 1: Two examples of sentence pairs.

3 System Overview

This section describes our method to Arabic WiC disambiguation based on two types of embeddings: CBOW Word2Vec and Lemma2Vec.

Given two sentences, s_1 and s_2 , and two words, v_i from s_1 and w_j from s_2 , the objective is to check whether v_i and w_j have the same meaning. To this end, our system extracts contexts of v_i and w_j from the sentence pair, represents them in two vectors and finally compares the two resulting vectors using the cosine similarity. The context of a word w of size n (denoted by $context(w, n)$) is composed of the words that surround the word w, with n words on the left and n words on the right (n varying between 1 and 10 in conducted experiments). To represent $context(w, n)$ in a vector space, two methods are proposed: first one is based on CBOW Word2Vec embeddings vectors (Mikolov et al., 2013) of the words appearing in the context, whereas the second is based on the Lemma2Vec of lemmas of words appearing in the context. To select the best way to represent the $context(w, n)$ by a vector, classification experiments were conducted using (i) different pooling operations, min, max, $mean$, and std to combine words/lemmas vectors of the context, (ii) different threshold values (between 0.55 and 0.85) and (iii) the removal of functional words (also called stop words). The later are used to express grammatical relationships among other words, they are characterized by they high frequency in the corpus which might affect the WiC disambiguation accuracy. The cosine similarity is then used to compare vectors of $context(v_i, n)$ and $context(w_j, n)$. Figure 2 illustrates how the cosine similarity is calculated from $context(v_i, 3)$ and $context(w_j, 3)$.

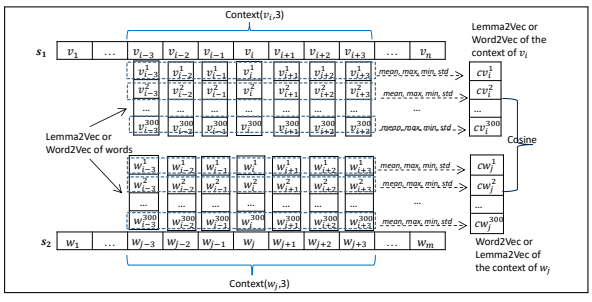


Figure 2: Calculation of $context(v_i, 3)$ and $context(w_j, 3)$ vectors and the cosine similarity between.

Classification experiments on SemEval-2021 `ar-ar` datasets were conducted using the following two CBOW Word2Vec models and two corresponding Lemma2Vec models: (i) Wiki-CBOW, a pretrained Word2Vec model from the set of AraVec models (Soliman et al., 2017) , (ii) our CBOW Word2Vec model that we trained ourselves, (iii) Lemma2Vec model that we constructed based on the Wiki-CBOW model, and (iv) Lemma2Vec that we constructed based on our CBOW Word2Vec model. Based on these four models, four experiments were conducted to tune the following parameters: context size ($context\_size$), $threshold$, pooling operation ($pooling$) and removing of functional words ($stop\_words$).

4 Corpus and trained Word Embeddings

Two CBOW Word2Vec models were used in our experiments. The Wiki-CBOW (Soliman et al., 2017), which consists of 234,173 vocabulary size, and another model we trained our self which consists of 334,161 vocabulary size. The Wiki-CBOW model was learnt from a corpus of Arabic Wikipedia articles of about 78 million words, the principal hyperparameters are: 5 for minimum word count and 5 for window size.

Our CBOW Word2Vec model was trained on Modern Standard Arabic corpora, such as (El-Khair, 2017; Abbas and Smaili, 2005; Abdelali et al., 2014) of about 3 billion words; it was fit using 300-dimensional word vectors, 100 the minimum count of words, training epochs of 5 and window size of 5.

Before training the Word2Vec model, several normalization and preprocessing steps were performed. First, all diacritics, punctuations, Madda character, digits (Hindi and Arabic), Latin characters (including accented letters) were removed. Second, some special Arabic letters are unified. Third, sequences of repeated characters with length larger than 2 were reduced to one character; repeated spaces were also replaced by one space. Fourth, different forms of Alifs (ٱ إ أ) are replaced with (ا). Spaces followed by a period character and new lines were considered to be end of sentence marks. The split method in Python is used in tokenization. The vocabulary size of the resulted model is 334,161.

5 Constructing the Lemma2Vec models

Two Lemma2Vec models were produced, based on both: the Wiki-CBOW Word2Vec model, and our CBOW Word2Vec model. Each vocabulary in

each of the Word2Vec models was lemmatized first. Then a vector for each lemma (*i.e.*, Lemma2Vec) is calculated as following: first all word forms belonging to this lemma are fetched, then their Word2Vec vectors are combined through a $mean$ pooling operation. The lemmatization process was performed using in-house tools and lexicographic databases [1] belonging to Birzeit University (Jarrar, 2021; Jarrar and Amayreh, 2019; Jarrar et al., 2019). In case of a word cannot be lemmatized due to misspelling, incorrect tokenization or in case of foreign word (not included in our database), then the corresponding Lemma2Vec is considered to be its Word2Vec vector.

Table 1 summarizes the lemmatization results that we performed on both, the Wiki-CBOW model and our CBOW Word2Vec model. The lemmatized words of SemEval-2021 all `ar-ar` dataset, as well as the Word2Vec and Lemma2Vec of `ar-ar` datasets words's vectors used in this paper are available on-line [2].

	Wiki-CBOW 78M words min_count 5	**Our Word2Vec** 3B words min_count 100
Unique word forms	234,173	334,161
Unique lemmas	100,040	54,788
Words not lemmatized	22,054	28,098

Table 1: Lemmatization results for both models.

6 Experiments Results and Discussion

Given our Arabic WiC disambigation method described in Section 3, and given the SemEval multilingual `dev.ar-ar` dataset provided by SemEval-2021 (Martelli et al., 2021), four classification experiments were conducted using the cosine similarity and based on the two Word2Vec models and the two Lemma2Vec models. The objective is to tune the following parameters for each model: $context\_size$ (ranging from 1 to 10), $threshold$ (we determined empirically the range from 0.55 to 0.85 with 0.1 step size), $pooling$ (min, max, $mean$ and std), and $stop\_words$ (yes, no). Then the values of parameters corresponding to the high F1-scores for TRUE (T) and FALSE (F) classes are selected in order to classify sentence pairs in the `test.ar-ar` dataset. For each model we did the following to find the high F1-scores for T and F: For each $context\_size$ (between 1 and 10) and for each value of the $stop\_words$ (yes or no) we plotted 8 line plots (4 for T and 4 for F) for each of the four pooling operations ($mean$, max, min and std) and for $threshold$ ranging from 0.55 to 0.85 (*i.e.*, 20 plots for each model, resulting 80 plots).

	Exp1		**Exp2**		**Exp3**		**Exp4**	
Model	Word2Vec Wiki-CBOW		Lemma2Vec Wiki-CBOW		Word2Vec our model		Lemma2Vec our model	
context_size	4		1		4		1	
pooling	*min*		*min*		*min*		*mean*	
threshold	0.66		0.56		0.83		0.58	
stop_words	*yes*		*yes*		*no*		*yes*	
Dataset	`dev.ar-ar`							
Tag	T	F	T	F	T	F	T	F
Precision	52	52	57	58	56	56	**56**	**56**
Recall	54	51	61	53	55	57	**55**	**58**
F1-score	53	52	59	56	55	56	**56**	**57**
Dataset	`test.ar-ar`							
Accuracy	57		59		59		**60**	

Table 2: Best F1-score, precision and recall values of the four experiments on `dev.ar-ar` dataset with the values of tuned parameters. Below are accuracies on `test.ar-ar` dataset.

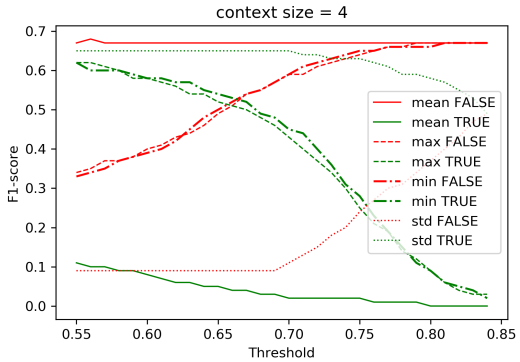
Figures 3a, 3b, 3c and 3d show the best 4 F1-scores line plots for each of the four models, and Table 2 shows the effective F1-scores values for T and F classes as well as precision and recall values (best results marked in bold). The values of parameters corresponding to the best result were then used in classifying the `test.ar-ar` dataset. The accuracies are reported in Table 2 as well.

As shown in Figure 3, the Lemma2Vec models have the tendency to perform better with shorter context sizes compared with the Word2Vec models. A possible reason may be that, in case of Lemma2Vec, the narrow meaning of words is affected due to the increase number of words involved in Lemma2Vec vector calculation. The impact of Lemma2Vec on the narrow meaning of words is discussed in the next subsection.

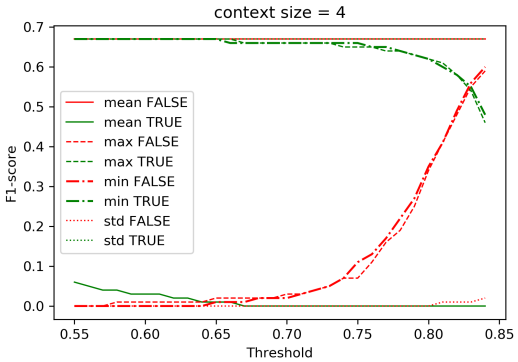
The results with yes for $stop\_words$ are slightly better but not significant. Additionally, the min pooling was generally the best operation to combine the context vectors, and the results of both min and max pooling were close to each other.

[1] `https://ontology.birzeit.edu`

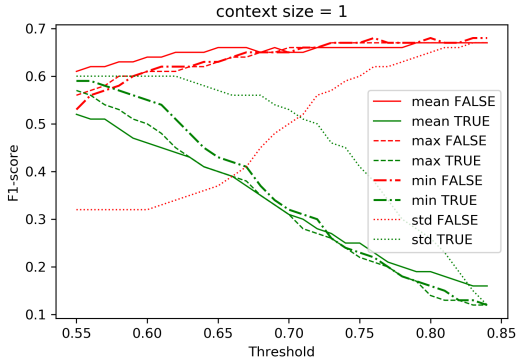
[2] `https://ontology.birzeit.edu/semeval2021_data.zip`



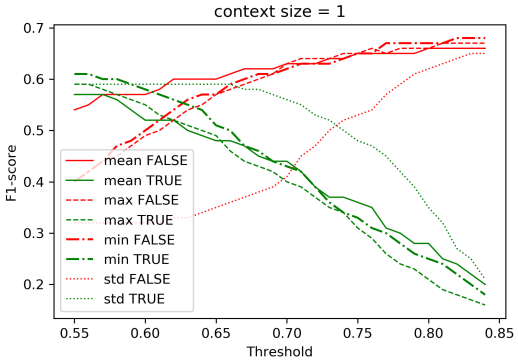
(a) **Wiki-CBOW Word2Vec** model.
$context\_size = 4$ - $pooling = min$
$threshold = 0.66$ - $stop\_words = yes$



(b) **our Word2Vec** model.
$context\_size = 4$ - $pooling = min$
$threshold = 0.83$ - $stop\_words = no$



(c) **Wiki-CBOW Lemma2Vec** model.
$context\_size = 1$ - $pooling = min$
$threshold = 0.56$ - $stop\_words = yes$



(d) **our Lemma2Vec** model.
$context\_size = 1$ - $pooling = mean$
$threshold = 0.58$ - $stop\_words = yes$

Figure 3: The best four F1-scores markers plots for each of the four models.The values of parameters are under each plot.

6.1 Lemma2Vec-Word2Vec Error Analyses

This subsection discusses the performance of using lemma-based vs. word-based models in the WiC disambiguation task, which we summarize in Table 3 and Table 4.

	TRUE	FALSE	Total
Correct L2V - Correct W2V	225	145	370
Correct L2V - Wrong W2V	118	98	216
Wrong L2V - Correct W2V	66	116	182
Wrong L2V - Wrong W2V	91	141	232
Total	500	500	1000

Table 3: Wiki-CBOW Lemma2Vec vs. Word2Vec

Table 3 presents the results of experiments 1 and 2 (using Word2Vec and Lemma2Vec of Wiki-CBOW) whereas Table 4 presents the results of experiments 3 and 4 (using Word2Vec and Lemma2Vec of our CBOW model). In each table, we compare between cases that were correctly or wrongly classified by both models. For example, the second row in Table 3 shows that 216 sentence pairs (118 TRUE class + 98 FALSE class) were correctly classified using the Wiki-CBOW's Lemma2Vec model but wrongly classified using the Word2Vec. Similarly, 182 sentence pairs in the third row were correctly classified using the Word2Vec but wrongly classified using the Lemma2Vec.

	TRUE	FALSE	Total
Correct L2V - Correct W2V	124	241	365
Correct L2V - Wrong W2V	188	45	233
Wrong L2V - Correct W2V	58	178	236
Wrong L2V - Wrong W2V	130	36	166
Total	500	500	1000

Table 4: Our Lemma2Vec vs. our Word2Vec

As shown in both tables' second and third rows, the Lemma2Vec did not significantly improve the overall results; but notably, the Lemma2Vec shows a significant improvement over Word2Vec for TRUE class whereas Word2Vec is better for FALSE class.

This conclusion is valid for all models, whatever are the corpora content, size and min_count

Figure 4: Example of errors.

hyperparameter.

To understand the gain and loss by the lemma-based models, we manually analyzed most cases. Figure 4 illustrates such cases. The first sentence pair in Figure 4 was correctly classified by the Lemma2Vec (in **Exp4**) and wrongly by the Word2Vec (in **Exp3**). This illustrates that the lemma vector as a generalized model for its inflections (*i.e.*, a mean of word forms' vectors) was better in deciding that both contexts are similar and that the two word forms have the same meaning. However, the second example in Figure 4 illustrates the other way. The Lemma2Vec was too general, and the Word2Vec was specific enough, to decide that the two word forms, in the two contexts, are different. The word from al-ğins (الجنس) could mean both *genus* and *sex*; however the other word form al-ʾağnās (الأجناس), is semantically distinctive by its own morphology - as it can only be plural of *genus*, and cannot be plural of *sex*.

To conclude, although Lemma2Vec outperforms Word2Vec in some cases (mostly in the TRUE sentence pairs class), it underperforms Word2Vec in others cases (mostly in the FALSE sentence pairs class). Since the distribution of TRUE and FALSE is equal in the datasets, the overall performance of both models is close to each other. Nevertheless, in case of an application scenario where a large proportion of sentence pairs is expected to be TRUE, we recommend the use of Lemma2Vec, otherwise the Word2Vec.

7 Conclusions and Further Work

We presented a set of experiments to evaluate the performance of using Word2Vec and Lemma2Vec models in Arabic WiC disambiguation, without using external resources or any context/sense embedding model. Different models were constructed based on two different corpora, and different types of parameters were tuned. The final results demonstrated that Lemma2Vec models are slightly better than Word2Vec models for Arabic WiC disambiguation. More specifically, we found that Lemma2Vec outperforms Word2Vec for TRUE sentence pairs, but underperforms it for FALSE sentence pairs.

We plan to extend our work to use our Lemma2Vec model to build a multi-prototype embeddings using the large lexicographic database available at Birzeit University. We plan also to fine tune the recently released Arabic BERT models, such as (Safaya et al., 2020; Antoun et al., 2020; Abdelali et al., 2021; Inoue et al., 2021), using the same database.

Acknowledgments

We would like to thank the shared task organizers and the reviewers for their valuable comments and efforts towards improving our manuscript. We would like to also thank Taymaa Hammouda for her technical support.

References

Mourad Abbas and Kamel Smaili. 2005. Comparison of topic identification methods for arabic language. In *Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP*, pages 14–17.

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In *LREC*, volume 14, pages 1044–1054.

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations.

Mohammad Khaled A Al-Maghasbeh and MP Bin Hamzah. 2015. Extract the semantic meaning of prepositions at arabic texts: an exploratory study. *Int J Comput Trends Technol*, 30(3):116–120.

Ali Alkhatlan, Jugal Kalita, and Ahmed Alhaddad. 2018. Word sense disambiguation for arabic exploiting arabic wordnet and word embedding. *Procedia computer science*, 142:50–60.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference*, page 9.

Nadia Bouhriz, Faouzia Benabbou, and EH Ben Lahmar. 2016. Word sense disambiguation approach for arabic text. *International Journal of Advanced Computer Science and Applications*, 7(4):381–385.

Ibrahim Bounhas, Raja Ayed, Bilel Elayeb, Fabrice Evrard, and Narjes Bellamine Ben Saoud. 2015. Experimenting a discriminative possibilistic classifier with reweighting model for arabic morphological disambiguation. *Computer Speech & Language*, 33(1):67–87.

KZ Bousmaha, S Charef Abdoun, L Hadrich Belguith, and MK Rahmouni. 2013. Une approche de désambiguïsation morpho_lexicale évaluée sur l'analyseur morphologique alkhalil. *Revue RIST—Vol*, 20(2):33.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ibrahim Abu El-Khair. 2017. Abu el-khair corpus: A modern standard arabic corpus. *International Journal of Recent Trends in Engineering & Research (IJRTER)*, 03(1):95–100.

Bilel Elayeb. 2019. Arabic word sense disambiguation: a review. *Artificial Intelligence Review*, 52(4):2475–2532.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Mustafa Jarrar. 2021. The arabic ontology - an arabic wordnet with ontologically clean content. *Applied Ontology Journal*, 16(1):1–26.

Mustafa Jarrar and Hamzeh Amayreh. 2019. An arabic-multilingual database with a lexicographic search engine. In *International Conference on Applications of Natural Language to Information Systems*, volume 11608 of *LNCS*, pages 234–246. Springer.

Mustafa Jarrar, Hamzeh Amayreh, and John P McCrae. 2019. Representing arabic lexicons in lemon - a preliminary study. In *The 2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 2402, pages 29–33. CEUR.

Mustafa Jarrar, Fadi Zaraket, Rami Asia, and Hamzeh Amayreh. 2018. Diacritic-based matching of arabic words. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(2):10:1–10:21.

Dan Klein, Kristina Toutanova, H Tolga Ilhan, Sepandar D Kamvar, and Christopher D Manning. 2002. Combining heterogeneous classifiers for word sense disambiguation. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*, pages 74–80.

Rim Laatar, Chafik Aloulou, and Lamia Hadrich Belguith. 2017. Word sense disambiguation of arabic language with word embeddings as part of the creation of a historical dictionary. In *LPKM*.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 41–48.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2016. Embedding words and senses together via joint knowledge-enhanced training. *arXiv preprint arXiv:1612.02703*.

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 2021. SemEval-2021 Task 2: Multilingual and Cross-lingual Word-in-Context Disambiguation (MCL-WiC). In *Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021)*.

Larousi Merhbene, Anis Zouaghi, and Mounir Zrigui. 2014. Approche basée sur les arbres sémantiques pour la désambiguïsation lexicale de la langue arabe en utilisant une procédure de vote. In *Proceedings of the 21st conference on natural language processing (TALN 2014), Marseille, France*, pages 281–290.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Maria Pelevina, Nikolay Arefyev, Chris Biemann, and Alexander Panchenko. 2017. Making sense of word embeddings. *arXiv preprint arXiv:1708.03390*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059,

Barcelona (online). International Committee for Computational Linguistics.

Rana Aref Salama, Abdou Youssef, and Aly Fahmy. 2018. Morphological word embedding for arabic. *Procedia computer science*, 142:83–93.

Pamela Shapiro and Kevin Duh. 2018. Morphological word embeddings for arabic neural machine translation in low-resource settings. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 1–11.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Nadia Soudani, Ibrahim Bounhas, Bilel ElAyeb, and Yahya Slimani. 2014. Generic normalization approach of arabic dictionaries for arabic word sense disambiguation. *Proceedings of Cinquième Journées Francophones sur les Ontologies (JFO), Hammamet, Tunisia*, pages 309–315.