A New Arabic Word Embedding model for Word Sense Induction

Abstract. We describe in this paper a new Arabic word embedding model for word sense induction. Word embedding models are gaining a great interest from the NLP research community and Word2vec is undoubtedly the most influential among these models. These models map all the words of the vocabulary to a vector space and then provide a semantic description of the words of a corpus as numerical vectors. Nevertheless, a well-known problem of these models is that they cannot handle polysemy. We present a new simple model for Arabic Word embedding which we experiment for the unsupervised task of word sense induction. The model is developed using Gensim tools for both SKIP-Gram and CBOW. Then the model allows the building of an indexer based on the cosine similarity using Annoy indexer which is faster than the Gensim similarity function. An Ego-network is used to study the structure of an individual's relationships and allows to build a graph of related words from the local neighbors. The different senses of the words are generated by clustering the graph. We have worked with two different news corpora: OSAC and Aracorpus. We have experimented the different models of the existing Aravec and our models to word sense induction and we obtained promising results. Our model shows good performance of word sense discrimination for a sample of Arabic ambiguous words.

Keywords: Word Embeddings, Word2Vec, Word Sense Induction, Arabic Language, NLP.

1 Introduction

A word sense is a discrete representation of one aspect of the meaning of a word, and then word senses are the set of the possible meanings of a given word that we can found in machine readable dictionaries, corpora, etc. [17] The choice of how to represent word senses is a fundamental problem in NLP and depends on the type of NLP application. the Sense inventory can be built in different ways: it is usually a fixed list of the senses of each word [15] [18]. The construction of handcrafted lexical resources or manually annotated data is expensive and time-consuming. Word Sense Induction (WSI) overcomes this problem by using clustering algorithms which do not need training data [16]. WSI is an open problem in NLP, related to word sense disambiguation WSD, which aims to automatically induce senses of words from a corpus. the corpus size has an important impact for WSI, however, clustering in a high dimensional text is a hard problem.

Word embeddings constitute an efficient method to represent words in a reduced dimension, they use a one-dimensional vector to represent words [2]. These models allow words with similar meaning to have a similar representation. However, these representations using a single vector are unable to capture multiple senses. In order to be able to benefit from word embedding techniques for individual word senses, several approaches have been proposed to relieve this issue [3] [4] [5] [19] [20] [21].

The contribution of this paper is a technique that automatically produces an Arabic sense inventory using Word Sense Induction via word embeddings, where word senses of the inventory are represented by word clusters.

To our knowledge, this is the first attempt to build automatically an Arabic sense inventory using word embeddings. Experiments show that our approach is promising and demonstrates good performance of word sense induction for a sample of Arabic ambiguous words.

2 Word Embedding Models

Word Embedding is one of the latest proposed solutions which encountered a great success, it has been proposed for the first time in 2003 by Bengio et al [1], and became popular with Word2Vec in 2013 [2]. These models map words into real-valued vectors in a low dimensional semantic space that can be learned by machine learning algorithms to make prediction of words and not counting words. The main advantage of these models, besides the low dimensionality, is that they can capture the information of words similarity; similar words have similar vectors. However, these models do not take into consideration lexical ambiguities, they represent all the senses of a word by a single vector representation [3]. In order to be able to benefit from word embedding techniques for individuals word senses, we automatically induce the different senses of Arabic Words and build sense inventory that can be used later for applications such as WSD.

2.1 Data Resources

The main goal of this work is to build a new Arabic word embedding model for word sense discrimination. To this end, we built both Skip-gram and CBOW Word2Vec models using two corpuses Open Source Arabic Corpus (OSAC) and Arabic Modern Standard Corpus named: AraCorpus; We then performed WSI with our obtained models and compared results with WSI obtained with the existing AraVec models.

The Open Source Arabic Corpus (OSAC). It is a corpus built from multiple Websites. It is divided into three main groups: BBC-Arabic Corpus which contains 1,860,786 (1.8M) words and 106,733 unique words after stop words removal, CNN-Arabic Corpus which contains 2,241,348 (2.2M) words and 144,460 unique words after stop words removal. Then OSAC collected from multiple websites presented in [6] which contains about 18,183,511 (18M) words and 449,600 unique words after stop words removal [6] [22]. We have not used the CNN-Arabic Corpus, because of problems of codification in the corpus.

The Arabic Modern Standard Corpus (AraCorpus). It is a collection of Arabic newspapers articles from ten Arabic countries. It has 102,134 articles, with 113 million words (800MB) and 296570 unique words [7] [22].

AraVec. It is a pre-trained distributed word representation open source project, it is free to use and offers powerful word embedding models. AraVec provides six different models built for three different Arabic corpora: Twitter, Wikipedia and from WWW pages. The Twitter corpus has 1090 million words and 164077 unique words. The Wikipedia corpus has 78.9 million words and 140319 unique words, and the WWW corpus has 2225.3 million words with 146273 unique words [8].

2.2 Pre-processing

To build a Word2Vec model, a pre-processing step is required. We use the Gensim tool developed by Radim Rehurek [9], which expects a sequence of sentences as input, where each sentence contains a list of words and each line in the file is a sentence.

AraCorpus is ready to use to build a word2vec model with genism, we just need to remove some special signs, but, OSAC Corpus needs further preprocessing such as normalization and removal of:

- Non- Arabic letter; like BBC Arabic or CNN Arabic in the beginning of each file in the corpus,
- Some special signs attached to the words like بحسم".
- Numbers,
- Vocalization like in: اطلاعاً
- Letters elongation

2.3 Learning a Word2vec model

After the preparation of the corpus, we built the CBOW and Skip-gram models using the Gensim toolkit for OSAC and AraCorpus. The AraVec models [8] were also built using the Gensim toolkit, which allows us to make a reasonable comparison between the obtained models and the AraVec models.

The Choice of the training parameters is an important step here. We selected a set of parameters according to prior evaluations of AraVec and experiments presented in [5]. We trained OSAC and AraCorpus Word Embedding models with 300 dimensions, context window size of 5, minimum word frequency of 5 and 4 threads. Table 1 shows the configuration used to build our models for OSAC and AraCorpus and the configuration used by the creators of AraVec [8].

Table 1. Word Embedding Models Configuration

Model Name	#unique word	Min Word Count	Window size	technique	time
OSAC-CBOW	140658	5	5	CBOW	751.5s
OSAC-SG				SG	660.5s
AraCorpus-CBOW	296570	5	5	CBOW	6340.7s
AraCorpus-SG				SG	5395.8s
Twitter-CBOW	164077	500	3	CBOW	1.5days
Twitter-SG				SG	
WWW-CBOW	146273	500	5	CBOW	4 days
WWW-SG				SG	
Wiki-CBOW	140319	20	5	CBOW	10 Hours
Wiki-SG				SG	

OSAC and AraCorpus Models were trained on a core™ i7-3632QM CPU 2.20GH with 8GB of RAM running Windows10 Pro, and AraVec models [8] were trained on a Quad-core Intel i7-3770 @3.4 GHz PC with 32 GB of RAM running Ubuntu 16.04.

3 Arabic Sense Induction using Word2vec models

We induce the Arabic sense inventory by clustering word similarity graph similarly to [5] [10] [13] [14], where a word sense is represented by a word cluster. For instance, the word « ذكر » with the sense « ذكر :mention أورد » can be represented by the cluster: اقوالا, واورد, ذم, اورد, ذكر, حكى.

To induce senses, we simply build an annoy indexer for a word embedding model to use as a graph of similar words for the vocabulary, then we generate an ego-network for any word in the vocabulary of the model, we built a graph of connected words then we can perform graph clustering on the graph of connected words.

3.1 Building a Word Similarity graph

The Word Similarity graph contains all words of the vocabulary as nodes linked by edges weighted by the Cosine Similarity between them, the graph is undirected. To build the graph we need to retrieve for each word in the vocabulary the k nearest neighbors, and present them in a file which consists of line of tuples of words with their similarity weight. We use the Annoy[1] (Approximate Nearest Neighbors Oh Yeah) library for similarity queries because the current implementation of the k nearest neighbor in vector space via genism has a linear complexity via brute force in the number of indexed documents although with extremely low constant factors while Annoy can find approximate nearest neighbors much faster. Annoy has the ability to use static files as indexes and this is an important feature that will help us later. The similarity between

[1] https://markroxor.github.io/gensim/static/notebooks/annoytutorial.html

two words $word_1$ and $word_2$is computed with the cosine similarity of the vector embedder of word1 and the vector embedded of word2, the formula is defined as follow:

$$cos\_sim_{w2v}\,(word_1, word_2) = \frac{word_1 \bullet word_2}{\parallel word_1 \parallel \bullet \parallel word_2 \parallel} (1)$$

Where $word_1$ and $word_2$ are word embeddings for $word_1$ and $word_2$. The choice of the number of nearest neighbors is motivated by prior studies [5] [11] [14].

Building an ego network. The Graph of the whole vocabulary may tell us some interesting things about the entire population and its sub-population but it does not tell us a lot about the opportunities and constraints facing individuals [12]. To induce senses for each word in the word similarity graph, we need to look closer to each word as an individual and its neighbors. This is possible with an ego-network where a single ego represents the individual word, the alters represent the neighbors of the word and the edges among those alters [12] [14] [5]. As we see in Fig.1shows the ego- network of “ذكر”: the ego is “ذكر”, the alters are: {ذكر، اتقاكم، اكرمكم، وانثى، اقوالا، ذم، وأورد، أورد، حكى} weighted with the cosine similarity distance. We use the provided index files that we mentioned in section 3.1 as a graph to build the ego networks from this index.

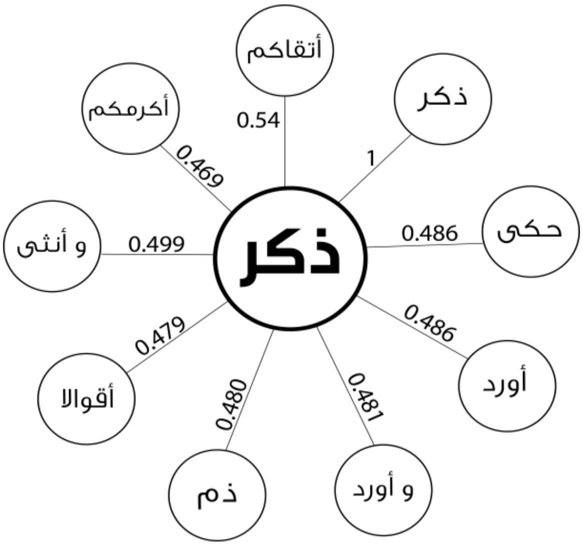

Fig. 1. Ego network for the word “ذكر” from the OSAC_CBOW Model for the 9 nearest neighbors

3.2 Word Sense Induction

To discriminate the senses of a given word W, we cluster the graph of connected words using the Chinese Whispers algorithm similarly to [5] [10] [14], each cluster for a given word represents a sense. Table 2 shows an instance of the results of induction for the word “ذكر”, the word is induced for two clusters (i.e. two senses) using the OSAC_CBOW model. The first cluster {حكي، أورد، وأورد، ذم، اقولا} represents the sense “mention/اورد” while the second cluster {اتقاكم، اكرمكم، وانثى} represents the sense “جنس/gender”.

Table 2. clustering of the neighbors of the word “ذكر” into two clusters representing two different senses (gender and mention)

ذكر1	0.486:حكى	0.486:اورد	0.481:واورد	0.480:ذم	0.479:اقوالا
ذكر2	0.499:وانثى	0.469:اكرمكم	0.454:اتقاكم		

The construction of the graph of connected words is based on the idea of relating two neighbors of a word, if one of them is one of the 200 nearest neighbors for the other word. The Algorithm 1. outlines the process of word sense induction, where the input `w2v_model` is one of the ten trained word embedding models and `AnnoyIndexer_of_w2v` indexes the embedded model got with the Annoy Indexer. Our algorithm is a variant of the WSI algorithm described in [5] where, we use the indexes data as word similarity graph which shows to be faster and easier.

Algorithm 1. Word Sense Induction

```
Input:  w2v_model, AnnoyIndexer_of_w2v
Output: sense inventory file for the words in
        the vocabulary of w2v_model
For each word′ in the vocabulary:
  G ← empty graph for connected words
  N ← 200 nearest neighbors of word′
      from annoyIndexer_of_w2v
   For each n ∊ N:
       NN ← 200 nearest neighbors of n
            from annoyIndexer_of_w2v
       For each nn ∊ NN:
          If nn ∊ N :
                 add_edge(nn,n,'weight'= W)

   chinese_whispers(G)
```

We calculate the weight W using four equations:

$$W = sim(n, nn) \tag{2}$$

$$W = (\mathrm{sim}\ (\mathrm{word}', \mathrm{nn}) + \mathrm{sim}\ (\mathrm{n}, \mathrm{nn}))/2 \tag{3}$$

$$W = (\mathrm{sim}\ (\mathrm{word}', \mathrm{nn}) + \mathrm{sim}\ (\mathrm{word}', \mathrm{n}) + \mathrm{sim}\ (\mathrm{n}, \mathrm{nn}))/3 \tag{4}$$

$$W{=}sim\ (word', nn) \tag{5}$$

The choice of this parameter has a big influence on the results of clustering. Table 3 Shows our evaluation of the granularity of the senses inventories given by using the fourth equation described previously. Where we note: "v-F-grained" mean "very fine-grained sense", "F-grained" mean "fine-grained sense", "C-grained" mean "coarse-grained sense" and "v-C-grained" mean "very coarse-grained sense".

For clustering, we used the Chinese Whispers algorithm [10] because it is parameter-free, thus we make no assumption about the number of word senses.

Table 3. granularity of senses obtained applying the four equations for the ten models

	Eq2.	Eq3.	Eq4.	Eq5.
Osac_CBOW	v-F-grained	v-F-grained	v-F-grained	C-grained
Osac_SG	v-F-grained	v-F-grained	v-F-grained	C-grained
Aracorpus_C	v-F-grained	F-grained	F-grained	C-grained
Aracorpus_S	v-F-grained	F-grained	F-grained	C-grained
Twr_CBOW	v-F-grained	v-F-grained	v-C-grained	C-grained
Twr_SG	v-F-grained	v-F-grained	v-C-grained	C-grained
Wiki_CBOW	v-F-grained	v-F-grained	v-C-grained	C-grained
Wiki_SG	v-F-grained	v-F-grained	v-C-grained	C-grained
WWW_CBOW	v-F-grained	F-grained	v-C-grained	C-grained
WWW_SG	v-F-grained	F-grained	v-C-grained	C-grained

4 Evaluation

In order to evaluate the approach presented in this paper, we will use our own judgement of what we have obtained, that because, for Arabic we do not know any method of evaluation, and we can't compute the precision and the recall of the proposed approach because the gold standard file of the Arabic language is not released yet.

We have built a sense inventory for the first 1000 words in each embedded model, then we chose two words that have more than one sense "العربية" and "العالم": the construction duration of the sense inventory for the ten models varies from 25-minute minimum to 40-minute maximum. We compared the results for only 6 models (OSAC, AraCorpus, and WWW of the AraVec for both CBOW and Skip-Gram models) because the nature of the three corpora is more similar comparing to Wikipedia or a Twitter corpus.

Table 4 Shows the induced senses results for the word "العربية" and "العالم", C refers to CBOW model and S for Skip-Gram model. We see the difference in results obtained between CBOW and SG models for the same corpus, there is no preference between them. In the WWW models, authors filtering the last "ة" to "ه", so we search the word « العربيه » instead of « العربية».

The examples show that our approach performs well with some words and some models, and for the other models, it tends to bring the different senses together or to induce several senses of a word even if all the clusters of the word express the same sense. Our results are promising and they depend on the quality and the nature of the corpus.

Table 4. Some examples show the induced senses for two Arabic words “العربية” and “العالم”, we show only the first four words in each cluster

		العربية	Sense

osac	C	{0.584:والاسلامية,0.510:تناولناها,0.498:الجماهرية,0.49:والافريقية}
		{0.522:اللغة,0.516:والعربية,0.507:والاردية,0.506:والسريانية}
			Nation
			Ar. Language

	S	{ 0.569:الخليجية, 0.556:الانجليزية,0.538:الانكليزية,0.529:الاوربية }	Ar. Language or Arabic Civilization
www	C	{ 0.553:المغاربيه, 0.545:الفارسيه , 0.544:النوبيه, 0.543:الكورديه}	Ar.Language

	S	{0.499:والخليجيه, 0.495:والاسلاميه, 0.494:والانجليزيه, 0.488:والاجنبيه}
		{0.472:تركيامعاهد, 0.466:تركيالمدارس, 0.465:تركيالمطاعم, 0.464:وبولفارا}
			Ar. Language
			?

Ara Cor	C	{0.555:العربية, 0.550:والاوردية, 0.545:اليكسو, 0.533:واللاتينية}	Ar.Language
	S	{0.575:الخليجية, 0.536:العربيبة, 0.523:الافريقية, 0.521:المغاربية}	Ar.Language or Region
		العالم	
osac	C	{ 0.501:بالعالم, 0.500:للعالم, 0.496:نهاييات, 0.495:ويسعي}	World cup

	S	{ 0.527:القارات, 0.522:الخليج, 0.509:اوروبا , 0.491:المعمورة}
		{0.543:عالمنا, 0.539:بالعالم, 0.524:والعالم, 0.522:للعالم}
			Geographic world
			The World

www	C	{ 0.582:بالعالم, 0.552:عالمنا, 0.551:اوروبا, 0.525:والعالم}	Geographic world

	S	{0.524:انحاء, 0.481:بالعالم, 0.470:الملككاس, 0.463:بالبرازيل}
		{0.453:اندريا, 0.430:ريكي}
			Geographic world

Ara Cor	C	{0.565:العربي, 0.559:الاسلامي, 0.538:لتخرجنا, 0.537:مستضافا}
		{0.562:للشاطيية, 0.547:للسوبربايك, 0.539:للوسطيات, 0.532:لراليات}
			The political world
			?

	S	{0.585:عالمنا, 0.513:للعالم, 0.499:اوروبا, 0.499:القارات}
		{0.437:اميركا, 0.437:مصر, 0.429:العراق, 0.427:الصين}
			Geographic world
			The political world or Geographic world

5 Conclusion

We presented in this paper a New Arabic word embedding models and a technic to use them to produce automatically Arabic senses inventories. First, we have presented how to build Arabic embedding models using available Arabic corpora (OSAC and AraCorpus); then, we described how to use the embedding models to induce senses for any word in the vocabulary by clustering the graph of connected words using Chinese Whispers algorithm. The construction of the graph of connected words for a given word W is based on the idea of relating two neighbors of a word W if one of them is one of the K-nearest neighbors for the other word. We get the k-nearest neighbors using the annoy indexer which can find approximate nearest neighbors faster than the genism similarity function.

Our results are promising, we can observe that the choice of the corpora and the preprocessing are two important steps, at this stage, we cannot say which of the models CBOW or Skip-gram is better to induce Arabic senses, however, the use of the two models together may give better results.

In future work, we would like to experiment our approach varying the nature of the corpora and filtering the Arabic Stop List words, and, apply these results to enhance other problems relying on word sense inventories.

References

1. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin, “A neural probabilistic language model”, The Journal of Machine Learning Research, 2003.
2. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient estimation of word representations in vector space”, CoRR, (2013)
3. Dai Quoc Nguyen, Dat Quoc Nguyen, Ashutosh Modi, Stefan Thater, Manfred Pinkal “A Mixture Model for Learning Multi-Sense Word Embeddings” Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017), pages 121–127, Vancouver, Canada, August 3-4, 2017.
4. Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. “Embeddings for word sense disambiguation: An evaluation study”. 2016
5. Pelevina Maria , Arefiev Nikolay , Biemann Chris , Panchenko Alexander ," Making Sense of Word Embeddings" Proceedings of the 1st Workshop on Representation Learning for NLP, August, 2016
6. Motaz K. Saad and Wesam Ashour, "OSAC: Open Source Arabic Corpus", 6th ArchEng International Symposiums, EEECS’10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, European University of Lefke, Cyprus, 2010.
7. Abdelali, A., Cowie, J., &Soliman, H. “Building a modern standard Arabic corpus.” Paper presented at the workshop on computational modeling of lexical acquisition, the split meeting. Croatia, 25-28 July 2005.
8. Abu Bakr Soliman, Kareem Eissa, Samhaa R. El-Beltagy, " AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP ",3rd International Conference on Arabic Computational Linguistics, ACLing 2017, Dubai, United Arab Emirates,5-6 November 2017.
9. R. Rehurek and P. Sojka, “Software framework for topic modelling with large corpora,” in In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010.
10. Chris Biemann, “Chinese Whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems”. In Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, pages 73–80, New York City, USA, 2006.
11. Alexander Panchenko. “Similarity measures for semantic relation extraction.” Ph.D. thesis, Universite catholique de Louvain, Louvain-la-Neuve, Belgium, 2013.

12. Hanneman, Robert A. and Mark Riddle. Introduction to social network methods. Riverside, CA: University of California, Riverside (published in digital form at http://faculty.ucr.edu/~hanneman/), 2005.
13. Chris Biemann. "Turk Bootstrap Word Sense Inventory 2.0: A Large-Scale Resource for Lexical Substitution". In Proceedings of the 8th International Conference on Language Resources and Evaluation, pages 4038–4042, Istanbul, Turkey.2012.
14. Alexander Panchenkoz, Eugen Ruppertz, Stefano Faralliy, Simone Paolo, Ponzettoy and Chris Biemannz "Unsupervised Does Not Mean Uninterpretable: The Case for Word Sense Induction and Disambiguation'‘, EACL,2017.
15. Kwong O.Y. "Word Senses and Problem Definition." In: New Perspectives on Computational and Cognitive Strategies for Word Sense Disambiguation. Springer Briefs in Electrical and Computer Engineering. Springer, New York,2013.
16. David Pinto, Paolo Rosso, Héctor Jiménez-Salazar, UPV-SI: word sense induction using self-term expansion, Proceedings of the 4th International Workshop on Semantic Evaluations, p.430-433, June 23-24, Prague, Czech Republic, 2007.
17. Daniel Jurafsky, James H. Martin,"Speech and Language Processing.", chapter 17, Draft of November 7, 2016.
18. Marek Kozlowski, Henryk Rybinski,"Word Sense Induction with Closed Frequent Termsets", Computational Intelligence, Volume 33, Number 3, 2017.
19. Anne Cocos, Marianna Apidianaki,and Chris Callison-Burch, "Word Sense Filtering Improves Embedding-Based Lexical Substitution", Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their applications, pages 110–119, Valencia, Spain, April 4 2017.
20. S Bartunov, D Kondrashkin, A Osokin, D Vetrov,"Breaking sticks and ambiguities with adaptive skip-gram", Artificial Intelligence and Statistics, 130-138, 2016.
21. Alexander Panchenko,"Best of Both Worlds: Making Word Sense Embeddings Interpretable", 10th edition of the Language Resources and Evaluation Conference, Portorož, Slovenia, 2016
22. Ibrahim Abu El-Khair," Abu El-Khair Corpus: A Modern Standard Arabic Corpus", International Journal of Recent Trends in Engineering & Research (IJRTER) Volume 02, Issue 11; November- 2016.