Semantic Similarity of Arabic Sentences with Word Embeddings

El Moatez Billah Nagoudi
LIM - Laboratoire d'Informatique et de Mathématiques, Université Amar Telidji de Laghouat, Algérie
e.nagoudi@lagh-univ.dz

Didier Schwab
LIG-GETALP
Univ. Grenoble Alpes
France
didier.schwab@imag.fr

Abstract

Semantic textual similarity is the basis of countless applications and plays an important role in diverse areas, such as information retrieval, plagiarism detection, information extraction and machine translation. This article proposes an innovative word embedding-based system devoted to calculate the semantic similarity in Arabic sentences. The main idea is to exploit vectors as word representations in a multidimensional space in order to capture the semantic and syntactic properties of words. IDF weighting and Part-of-Speech tagging are applied on the examined sentences to support the identification of words that are highly descriptive in each sentence. The performance of our proposed system is confirmed through the Pearson correlation between our assigned semantic similarity scores and human judgments.

Keywords: Semantic Sentences Similarity, Word Embedding, Word Representations, Space Vector Model.

1 Introduction

Text Similarity is an important task in several application fields, such as information retrieval, plagiarism detection, machine translation, topic detection, text classification, text summarization and others. Finding similarity between two texts, paragraphs or sentences, is based on measuring, directly or indirectly, the similarity between words.

There are two known types of words similarity: lexical and semantic. The first one handles the words as a stream of characters: words are similar lexically if they share the same characters in the same order (Manning et al., 2008). There are many techniques of lexical similarity measures, the most known are : Damerau-Levenshtein (Levenshtein, 1966), Needleman Wunsch (Needleman and Wunsch, 1970), LCS (Chvatal and Sankoff, 1975), JaroWinkler (Winkler, 1999), etc.

The second type aims to quantify the degree to which two words are semantically related. As an example they can be, synonyms, represent the same thing or they are used in the same context. The classical way to measure this semantic similarity is by using linguistic resources, like WordNet (Miller, 1995), HowNet (Dong and Dong, 2003), BabelNet (Navigli and Ponzetto, 2012) or Dbnary (Sérasset, 2015). However, the word embedding techniques can be a more effective alternative to these linguistic databases (Mikolov et al., 2013a).

In this article we focus our investigation on measuring the semantic similarity between short Arabic sentences using word embedding representations. We also consider the IDF weighting and Part-of-Speech tagging techniques in order to improve the identification of words that are highly descriptive in each sentence.

The rest of this article is organized as follows, the next section describes work related to word representations in vector space. In Section 3, we present three variants of our proposed word embedding-based system. Section 4 describes the experimental results of this study. Finally, our conclusion and some future research directions are drawn in Section 5.

2 Word Embedding Models

Words representations as vectors in a multidimensional space allows to capture the semantic and syntactic properties of the language (Mikolov et al., 2013a). These representations can serve as a fundamental building unit to many applications of

Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP), pages 18–24,
Valencia, Spain, April 3, 2017.

Natural Language Processing (NLP). In the literature, several techniques are proposed to build vectorized space representations.

For instance, Collobert and Weston (2008) have proposed a unified system based on a deep neural network architecture, and trained jointly with many well known NLP tasks, including: Chunking, Part of Speech tagging, Named Entity Recognition and Semantic Role Labeling. Their word embedding model is stored in a matrix $M \in R^{d*|D|}$, where D is a dictionary of all unique words in the training data, and each word is embedded into a $d\text{-}dimensional$ vector. The sentences are represented using the embeddings of their forming words. A similar idea was independently proposed and used by Turian et al. (Turian et al., 2010).

Mnih and Hinton (2009) have proposed another form to represent words in vector space, named Hierarchical Log-Bilinear Model (HLBL). Like virtually all neural language models, the HLBL model represents each word with a real-valued feature vector. For n-gram word-based, HLBL concatenates the $n-1$ first embedding words $(w_1..w_{n-1})$ and learns a neural linear model to predicate the last word w_n.

Mikolov et al. (Mikolov et al., 2013c) have used a recurrent neural network (RNN) (Mikolov et al., 2010) to build a neural language model. The RNN encode the context word by word and predict the next word. The weights of the trained network are used as the words embeddings vectors.

Mikolov et al. (Mikolov et al., 2013a) (Mikolov et al., 2013b) have proposed two other approaches to build a words representations in vector space. using a simplified version of Bengio et al. (Bengio et al., 2003) neural language mode. They replaced the hidden layer by a simple projection layer in order to boost performance. In their work, two models are presented: the continuous bag-of-words model (CBOW) (Mikolov et al., 2013a), and the skip-gram model (SKIP-G) (Mikolov et al., 2013b).

In the first one, the continuous bag of word model CBOW (Mikolov et al., 2013a), predicts a pivot word according to the context by using a window of contextual words around it. Given a sequence of words $S = w_1, w_2, ..., w_i$, the CBOW model learns to predict all words w_k from their surrounding words $(w_{k-l}, ..., w_{k-1}, w_{k+1}, ..., w_{k+l})$. The second model SKIP-G, predicts surrounding words of the current pivot word w_k (Mikolov et al., 2013b).

Pennington et al. (Pennington et al., 2014) proposed a Global Vectors (GloVe) to build a words representations model, GloVe uses the global statistics of word-word co-occurrence to build co-occurrence matrix M. Then, M is used to calculate the probability of word w_i to appear in the context of another word w_j, this probability $P(i/j)$ represents the relationship between words.

3 System Description

3.1 Model Used

In (Mikolov et al., 2013a), all the methods (Collobert and Weston, 2008), (Turian et al., 2010), (Mnih and Hinton, 2009), (Mikolov et al., 2013c) have been evaluated and compared, and they show that CBOW and SKIP-G are significantly faster to train with better accuracy compared to these techniques. For this reason, we have used the CBOW word representations for Arabic model[1] proposed by Zahran et al. (Zahran et al., 2015). To train this model, they have used a large collection from different sources counting more than 5.8 billion words :

- Arabic Wikipedia (WikiAr, 2006).
- BBC and CNN Arabic corpus (Saad and Ashour, 2010).
- The open parallel corpus (Tiedemann, 2012).
- Arabase Corpus (Raafat et al., 2013).
- Osac: Open source arabic corpora. (Saad and Ashour, 2010)
- MultiUN corpus (Chen and Eisele, 2012)
- AGC Arabic Gigaword Corpus.
- King Saud University corpus (ksucorpus, 2012).
- Meedan Arabic corpus (Meedan, 2012).
- LDC Arabic newswire.
- Raw Quran text (Quran, 2007).
- KDE4 localization files (Tiedemann, 2009).
- Khaleej and Watan 2004 (Khaleej, 2004).

Training the Arabic CBOW model require choice of some parameters affecting the resulting vectors. All the parameters used by Zahran et al. (Zahran et al., 2015) are shown in Table 1.

[1] https://sites.google.com/site/mohazahran/data

The Arabic CBOW Model Parameters	
Parameter	**Value**
Vector size	300
Window	5
Sample	$1e-5$
Hierarchical Softmax	NO
Negative	10
Freq. thresh.	100

Table 1: Training configuration parameters

Where:

- **Vector size:** dimensionality of the word vectors.
- **Window:** number of words considered around the pivot word (context).
- **Sample:** threshold for sub-sampling of frequent words.
- **Hierarchical Softmax:** approximation of the full softmax used to predict words during training.
- **Negative:** number of negative examples in the training.
- **Frequency threshold:** threshold to discard less frequent words.

3.2 Words Similarity

We used CBOW model in order to identify the near matches between two words w_i and w_j (e.g. synonyms, singular, plural, feminization or closely related semantically). The similarity between w_i and w_j is obtained by comparing their vector representations v_i and v_j respectively. The similarity between v_i and v_j can be evaluated using the cosine similarity, euclidean distance, Manhattan distance or any other similarity measure functions. For example: let "الجامعة" (*university*), "المساء" (*evening*) and "الكلية" (*faculty*) be three words. The similarity between them is measured by computing the cosine similarity between their vectors as follows:

$$sim(\text{المساء}, \text{الجامعة}) = cos(V(\text{المساء}), V(\text{الجامعة})) = 0.13$$
$$sim(\text{الكلية}, \text{الجامعة}) = cos(V(\text{الجامعة}), V(\text{الكلية})) = 0.72$$

That means that, the words "الكلية" (*faculty*) and "الجامعة" (*university*) are semantically closer than "المساء" (*evening*) and "الجامعة" (*university*).

3.3 Sentences similarity

Let $S_1 = w_1, w_2, ..., w_i$ and $S_2 = w'_1, w'_2, ..., w'_j$ be two sentences, their word vectors are $(v_1, v_2, ..., v_i)$ and $(v'_1, v'_2, ..., v'_j)$ respectively. We have used three methods to measure the similarity between sentences. Figure 1 illustrates an overview of the procedure for computing the similarity between two candidate sentences in our system.

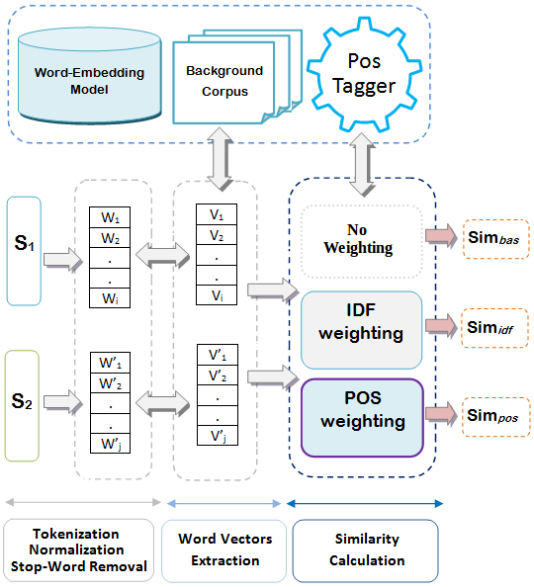

Figure 1: The architecture of the proposed system

In the following, we explain our proposed methods to compute the semantic similarity among sentences.

3.3.1 No Weighting Method

A simple way to compare two sentences, is to sum their words vectors. In addition, this method can be applied to any size of sentences. The similarity between S_1 and S_2 is obtained by calculating the cosine similarity between V_1 and V_2, where:

$$\begin{cases} V_1 = \sum_{k=1}^{i} v_k \\ V_2 = \sum_{k=1}^{j} v'_k \end{cases}$$

For example, let S_1 and S_2 be two sentences:

S_1 = "ذهب يوسف إلى الكلية" (*Joseph went to college*).

S_2 = "يوسف يمضى مسرعا للجامعة" (*Joseph goes quickly to university*).

The similarity between S1 and S2 is obtained as follows:

step 1: Sum of the word vectors

$V_1 = V(\text{الكلية}) + V(\text{يوسف}) + V(\text{ذهب})$

$V_2 = V(\text{للجامعة}) + V(\text{مسرعا}) + V(\text{يمضى}) + V(\text{يوسف})$

step 2: Calculate the similarity

The similarity between S_1 and S_2 is obtained by calculating the cosine similarity between V_1 and V_2.

$$sim(S_1, S_2) = cos(V_1, V_2) = 0.71$$

In order to improve the similarity results, we have used two weighting functions based on the Inverse Document Frequency IDF (Salton and Buckley, 1988) and the Part-Of-Speech tagging (POS tagging) (Schwab, 2005) (Lioma and Blanco, 2009).

3.3.2 IDF Weighting Method

In this variant, the Inverse Document Frequency IDF concept is used to produce a composite weight for each word in each sentence. The IDF weighting of words (Salton and Buckley, 1988) is traditionally used in information retrieval (Turney and Pantel, 2010) and can be employed in our system. The *idf weight* serves as a measure of how much information the word provides, that is, whether the term that occurs infrequently is good for discriminating between documents (in our case sentences).

This technique uses a large collection of document (background corpus), generally the same genre as the input corpus that is to be semantically verified. In order to compute the *idf weight* for each word, we have used the BBC and CNN Arabic corpus[2] (Saad and Ashour, 2010) as a background corpus. In fact, the idf of each word is determined by using the formula:

$$idf(w) = log(\frac{S}{WS})$$

where S is the total number of sentences in the corpus and WS is the number of sentences containing the word w. The similarity between S_1 and S_2 is obtained by calculating the cosine similarity between V_1 and V_2, $cos(V_1, V_2)$ where:

$$\begin{cases} V_1 = \sum_{k=1}^{i} idf(w_k) * v_k \\ V_2 = \sum_{k=1}^{j} idf(w'_k) * v'_k \end{cases}$$

and $idf(w_k)$ is the weight of the word w_k in the background corpus.

[2] https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/

Example: let's continue with the sentences of the previous example, and suppose that IDF weights of their words are:

ذهب	يوسف	الكلية	يمضى	مسرعا	الجامعة
0.27	0.37	0.31	0.29	0.22	0.34

step 1: Sum of vectors with IDF weights

$V_1 = V(\text{الكلية}) * 0.31 + V(\text{يوسف}) * 0.37 + V(\text{ذهب}) * 0.27$

$V_2 = V(\text{الجامعة}) * 0.34 + V(\text{مسرعا}) * 0.22 + V(\text{يمضى}) * 0.29$
$+ V(\text{يوسف}) * 0.37$

step 2: Calculate the similarity

The cosine similarity is applied to computed a similarity score between V_1 and V_2.

$$sim(S_1, S_2) = cos(V_1, V_2) = 0.78$$

We note that the similarity result between the two sentences is better than the previous method.

3.3.3 Part-of-speech weighting Method

An alternative technique is the application of the Part-of-Speech tagging (POS tag) for identification of words that are highly descriptive in each input sentence (Schwab, 2005) (Lioma and Blanco, 2009). For this purpose, we have used the POS tagger for Arabic language proposed by G. Braham et al. (Gahbiche-Braham et al., 2012) to estimate the part-of-speech of each word in sentence. Then, a weight is assigned for each type of tag in the sentence. For example, $verb = 0.4$, $noun = 0.5$, $adjective = 0.3$, $preposition = 0.1$, etc.

The similarity between S1 and S2 is obtained in three steps (Schwab, 2005) as follows:

step 1: POS tagging

In this step the POS tagger of G. Braham et al. (Gahbiche-Braham et al., 2012) is used to estimate the POS of each word in sentence.

$$\begin{cases} Pos\_tag(S_1) = Pos_{w_1}, Pos_{w_2}, ..., Pos_{w_i} \\ Pos\_tag(S_2) = Pos_{w'_1}, Pos_{w'_2}, ..., Pos_{w'_j} \end{cases}$$

The function $Pos\_tag(S_i)$ returns for each word w_k in S_i its estimated part of speech Pos_{w_k} .

step 2: POS weighting

At this point we should mention that, the weight of each part of speech can be fixed empirically. Indeed, we based on the training data of SemEval-

2017 (Task 1)[3] to fix the POS weights.

$$\begin{cases} V_1 &= \sum_{k=1}^{i} Pos\_weight(Pos_{w_k}) * v_k \\ V_2 &= \sum_{k=1}^{j} Pos\_weight(Pos_{w'_k}) * v'_k \end{cases}$$

where $Pos\_weight(Pos_{w_k})$ is the function which return the weight of POS tagging of w_k.

step 3: Calculate the similarity

Finally, the similarity between S_1 and S_2 is obtained by calculating the cosine similarity between V_1 and V_2 as follows:

$$sim(S_1, S_2) = cos(V_1, V_2)$$

Example: let us continue with the same example above.

S_1 = "ذهب يوسف إلى الكلية" (*Joseph went to college*).

S_2 = "يوسف يمضى مسرعا للجامعة" (*Joseph goes quickly to university*).

and suppose that POS weights are:

$verb$	$noun$	$noun\_prop$	adj	$prep$
0.4	0.5	0.7	0.3	0.1

step 1: Pos tagging

The function $Pos\_tag(S_i)$ is applied to each sentence.

$$\begin{cases} Pos\_tag(S_1) = verb\ noun\_prop\ noun \\ Pos\_tag(S_2) = noun\_prop\ verb\ adj\ noun \end{cases}$$

step 2: Sum of vectors with POS weighting

$V_1 = V(\text{الكلية}) * 0.5 + V(\text{يوسف}) * 0.7 + V(\text{ذهب}) * 0.4$

$V_2 = V(\text{الجامعة}) * 0.5 + V(\text{مسرعا}) * 0.3 + V(\text{يمضى}) * 0.4 + V(\text{يوسف}) * 0.7$

step 3: Calculate the similarity

$$sim(S_1, S_2) = cos(V_1, V_2) = 0.82$$

4 Experiments And Results

4.1 Test Sample

In order to measure effectively the performances of our system, a large collection are necessary. In fact, we have used a dataset of 750 pairs of sentences drawn from publicly Microsoft Research Video Description Corpus (MSR-Video) (MSR-video, 2016), and manually translated into Arabic. The sentence pairs have been manually tagged by four annotators, and the similarity score is the mean of the annotators. This score is a float number between "0" (indicating that the meaning of sentences are completely independent) to "1" (signifying meaning equivalence).

4.2 Preprocessing

In order to normalize the sentences for the semantic similarity step, a set of preprocessing are performed on the data set. All sentences went through by the following steps:

1. Stop-word removal.
2. Remove punctuation marks, diacritics and non letters.
3. We normalized أ ، إ ، آ to ا and ة to ه.
4. Replace final ى followed by ء with ئ.
5. Normalizing numerical digits to the token "Num".

4.3 Results

To evaluate the performance of our system, our three approaches were assessed based on their accuracy on the 750 sentences in the MSR-Video corpus. An example of our results is shown in Table 2.

Sentence Pair	Hum.	Methods		
		No Weig.	IDF	POS
ذهب يوسف إلى الكلية يوسف يمضى مسرعا للجامعة	0.90	0.71	0.78	0.82
إمرأة تتحدث على الهاتف صبيان يتحدثان على الهاتف	0.35	0.65	0.45	0.40
رجل يصب المكرونة في طبق المتسابق في سيارة الإسعاف	0.0	0.15	0.13	0.13
إمرأة تضع الماكياج إمرأة تضع المساحيق على وجهها	0.92	0.55	0.67	0.72
يزيل ترسبات السمكة رجل يزيل الترسبات من السمكة	1.0	0.85	0.92	0.94
كلب يقرأ كتابا للطفل يقرأ طفل كتابا عن الكلاب	0.20	0.82	0.87	0.88

Table 2: Example of sentence similarity results

The sentence pairs in Table 2, were selected randomly from our dataset. It can be seen that the similarity estimation provided by our system are fairly consistent with human judgements. How-

[3] http://alt.qcri.org/semeval2017/task1/data/uploads/

ever, the similarity score is not good enough when two sentences share the same words, but with a totally different meaning, like in the last pair of sentences.

On the other hand, we calculate the Pearson correlation between our assigned semantic similarity scores and human judgements. The results are presented in Table 3.

Approach	Correlation
Basic method	72.33 %
IDF-weighting method	78.20%
POS tagging method	79.69%

Table 3: Correlation results

These results indicate that when the no weighting method is used the correlation rate reached 72.33%. Both IDF-weighting and POS tagging approaches significantly outperformed the correlation to more than 78% (respectively 78.2% and 79.69%).

5 Conclusion and Future Work

In this article, we presented an innovative word embedding-based system to measure semantic relations between Arabic sentences. This system is based on the semantic properties of words included in the word-embedding model. In order to make further progress in the analysis of the semantic sentence similarity, this article showed how the IDF weighting and Part-of-Speech tagging are used to support the identification of words that are highly descriptive in each sentence. In the experiments we have shown how these techniques improve the correlation results. The performance of our proposed system was confirmed through the Pearson correlation between our assigned semantic similarity scores and human judgements. As future work, we can make more improvement in the semantic similarity results by a smart hybridisation between both IDF weighting and POS tagging techniques.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Yu Chen and Andreas Eisele. 2012. Multiun v2: Un documents with multilingual alignments. In *LREC*, pages 2500–2504.

Václáv Chvatal and David Sankoff. 1975. Longest common subsequences of two random sequences. *Journal of Applied Probability*, 12(02):306–315.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Zhendong Dong and Qiang Dong. 2003. Hownet-a hybrid language and knowledge resource. In *Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003 International Conference on*, pages 820–824. IEEE.

Souhir Gahbiche-Braham, Hélene Bonneau-Maynard, Thomas Lavergne, and François Yvon. 2012. Joint segmentation and pos tagging for arabic using a crf-based classifier. In *LREC*, pages 2107–2113.

Khaleej. 2004. Khaleej and watan corpus https://sites.google.com/site/mouradabbas9/corpora, (accessed january 20,2017).

ksucorpus. 2012. King saud university corpus, http://ksucorpus.ksu.edu.sa/ar/ (accessed january 20,2017).

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Christina Lioma and Roi Blanco. 2009. Part of speech based term weighting for information retrieval. In *European Conference on Information Retrieval*, pages 412–423. Springer.

Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

Meedan. 2012. Meedan's open source arabic english, https://github.com/anastaw/meedan-memory, (accessed january 20,2017).

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track*, pages 1301–3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1081–1088. Curran Associates, Inc.

MSR-video. 2016. Microsoft research video corpus, https://www.microsoft.com/en-us/download/details.aspx?id=52422, (accessed january 21,2017).

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Quran. 2007. Raw quran text, http://tanzil.net/ download/, (accessed january 20,2017).

Hazem M Raafat, Mohamed A Zahran, and Mohsen Rashwan. 2013. Arabase-a database combining different arabic resources with lexical and semantic information. In *KDIR/KMIS*, pages 233–240.

Motaz K Saad and Wesam Ashour. 2010. Osac: Open source arabic corpora. In *6th ArchEng Int. Symposiums, EEECS*, volume 10.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Didier Schwab. 2005. *Approche hybride-lexicale et thématique-pour la modélisation, la détection et lexploitation des fonctions lexicales en vue de lanalyse sémantique de texte*. Ph.D. thesis, Université Montpellier II.

Gilles Sérasset. 2015. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web*, 6(4):355–361.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Peter D Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188.

WikiAr. 2006. Arabic wikipedia corpus, http://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/, (accessed january 21,2017).

William E Winkler. 1999. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer.

Mohamed A Zahran, Ahmed Magooda, Ashraf Y Mahgoub, Hazem Raafat, Mohsen Rashwan, and Amir Atyia. 2015. Word representations in vector space and their applications for arabic. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 430–443. Springer.