LIM-LIG at SemEval-2017 Task1: Enhancing the Semantic Similarity for Arabic Sentences with Vectors Weighting

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

Jérémy Ferrero
Compilatio
276 rue du Mont Blanc
74540 Saint-Félix, France
LIG-GETALP
Univ. Grenoble Alpes, France
jeremy.ferrero@imag.fr

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

Abstract

This article describes our proposed system named *LIM-LIG*. This system is designed for SemEval 2017 Task1: Semantic Textual Similarity (Track1). *LIM-LIG* proposes an innovative enhancement to word embedding-based model devoted to measure the semantic similarity in Arabic sentences. The main idea is to exploit the word representations as vectors in a multidimensional space 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. *LIM-LIG* system achieves a Pearsons correlation of 0.74633, ranking 2nd among all participants in the Arabic monolingual pairs STS task organized within the SemEval 2017 evaluation campaign.

1 Introduction

Semantic Textual Similarity (STS) is an important task in several application fields, such as information retrieval, machine translation, plagiarism detection and others. STS measures the degree of similarity between the meanings of two text sequences (Agirre et al., 2015). Since SemEval 2013, STS has been one of the official shared tasks.

This is the first year in which SemEval has organized an Arabic monolingual pairs STS. The challenge in this task lies in the interpretation of the semantic similarity of two given Arabic sentences, with a continuous valued score ranging from 0 to 5. The Arabic STS measurement could be very useful for several areas, including: disguised plagiarism detection, word-sense disambiguation, latent semantic analysis (LSA) or paraphrase identification. A very important advantage of SemEval evaluation campaign, is enabling the evaluation of several different systems on a common datasets. Which makes it possible to produce a novel annotated datasets that can be used in future NLP research.

In this article we present our LIM-LIG system devoted to enhancing the semantic similarity between Arabic sentences. In STS task (Arabic monolingual pairs) SemEval 2017, the LIM-LIG system propose three methods to measure this similarity: No weighting, IDF weighting and Part-of-speech weighting Method. The best submitted method (Part-of-speech weighting) achieves a Pearsons correlation of 0.7463, ranking 2nd in the Arabic monolingual STS task. In addition, we have proposed another method (after the competition) named Mixed method, with this method, the correlation rate reached 0.7667, which represent the best score among the different submitted methods involved in the Arabic monolingual STS task.

2 Word Embedding Models

In the literature, several techniques are proposed to build word-embedding model.

For instance, Collobert and Weston (2008) have proposed a unified system based on a deep neural network architecture. 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. Mnih and Hinton (2009) have proposed the Hierarchical Log-Bilinear Model (HLBL). The HLBL Model 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. (2013a, 2013b) have proposed

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 134–138,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

two other approaches to build a words representations in vector space. The first one named 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.(2014) proposed a Global Vectors (GloVe) to build a words representations model, GloVe uses the global statistics of word-word co-occurrence 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. (2015). To train this model, they have used a large collection from different sources counting more than 5.8 billion words including: Arabic Wikipedia (WikiAr, 2006), BBC and CNN Arabic corpus (Saad and Ashour, 2010), Open parallel corpus (Tiedemann, 2012), Arabase Corpus (Raafat et al., 2013), Osac corpus (Saad and Ashour, 2010), MultiUN corpus (Chen and Eisele, 2012), KSU corpus (ksucorpus, 2012), Meedan Arabic corpus (Meedan, 2012) and other (see Zahran et al. 2015).

3.2 Words Similarity

We used CBOW model in order to identify the near matches between two words w_i and w_j. 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 words vectors representations are $(v_1, v_2, ..., v_i)$ and $(v'_1, v'_2, ..., v'_j)$ respectively. There exist several ways to compare two sentences. For this purpose, we have used four 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.

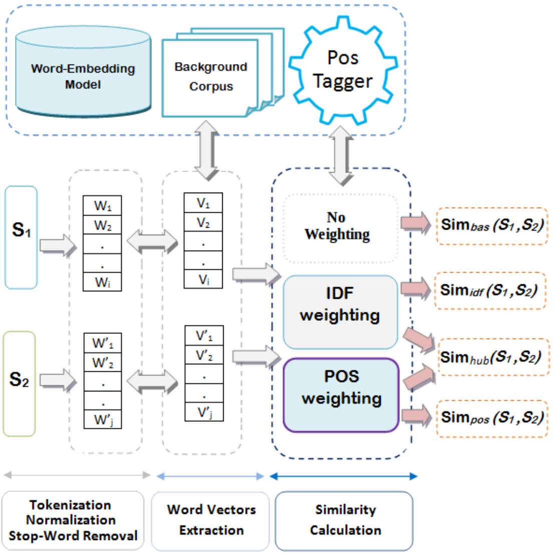

Figure 1: Architecture of the proposed system.

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:

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

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

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 S_1 and S_2 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 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.

Example: let us 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 (Lioma and Blanco, 2009). For this purpose, we have used the POS tagger for Arabic language proposed by G. 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 S_1 and S_2 is obtained in three steps (Ferrero et al., 2017) as follows:

Step 1: POS tagging

In this step the POS tagger of G. 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: $sim(S_1, S_2) = cos(V_1, V_2)$.

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

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

Example:
Let us continue with the same example, 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

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

3.3.4 Mixed weighting

We have proposed another method (after the competition), this method propose to use both IDF and the POS weightings simultaneously. The similarity between S_1 and S_2 is obtained as follows:

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

If we apply this method to the previous example, using the same weights in Section 3.2 and 3.3, we will have: $Sim(S_1, S_2) = Cos(V_1, V_2) = 0,87$.

4 Experiments And Results

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

4.2 Tests and Results

To evaluate the performance of our system, our four approaches were assessed based on their accuracy on the 250 sentences in the STS 2017 Monolingual Arabic Evaluation Sets v1.1[4]. We calculate the Pearson correlation between our assigned semantic similarity scores and human judgements. The results are presented in Table 1.

Approach	Correlation
Basic method (run 1)	0.5957
IDF-weighting method (run 2)	0.7309
POS tagging method (run 3)	0.7463
Mixed method	0.7667

Table 1: Correlation results

These results indicate that when the no weighting method is used the correlation rate reached 59.57%. Both IDF-weighting and POS tagging approaches significantly outperformed the correlation to more than 73% (respectively 73.09% and 74.63%). We noted that, the Mixed method achieve the best correlation (76.67%) of the different techniques involved in the Arabic monolingual pairs STS task.

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 are going to combine these methods with those of other classical techniques in NLP field such as: n-gram, fingerprint and linguistic resources.

Acknowledgments

This work was carried out as part of a PNE scholarship funded by Ministry of Higher Education and Scientific Research of Algeria, with an international collaboration between two research laboratories: LIM Laboratoire d'Informatique et de Mathmatiques Laghouat, Algeria and LIG Laboratoire d'Informatique de Grenoble (GETALP Team), France.

[4] http://alt.qcri.org/semeval2017/task1/data/uploads /sts2017.eval.v1.1.zip

References

Eneko Agirre, Carmen Baneab, Claire Cardiec, Daniel Cerd, Mona Diabe, Aitor Gonzalez-Agirrea, Weiwei Guof, Inigo Lopez-Gazpioa, Montse Maritxalara, Rada Mihalceab, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

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

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.

Jérémy Ferrero, Frédéric Agnès, Laurent Besacier, and Didier Schwab. 2017. Using Word Embedding for Cross-Language Plagiarism Detection. In *European Association for Computational Linguistics (EACL)*, Volume "short papers" EACL 2017, Valence, Spain, April.

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

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.

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

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.

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.

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.

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.

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