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

El Moatez Billah Nagoudi, Jérémy Ferrero, Didier Schwab

To cite this version:

HAL Id: hal-01531255
https://hal.archives-ouvertes.fr/hal-01531255
Submitted on 1 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 $M \in \mathbb{R}^{d \times |D|}$, where $D$ is a dictionary of all unique words in the training data, and each word is embedded into a $d$-dimensional vector. Mnih and Hinton (2009) have proposed the Hierarchical Log-Bilinear Model (HLBL). The HLBL Model concatenates the $n - 1$ first embedding words $(w_1, \ldots, w_{n-1})$ and learns a neural linear model to predicate the last word $w_n$.

Mikolov et al. (2013a, 2013b) have proposed
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 \( (w_{k-1}, w_{k-2}, ..., w_{k+l}) \). 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 (ksu-corpus, 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), "الكلية" (faculty) and "ال مساء" (evening) be three words. The similarity between them is measured by computing the cosine similarity between their vectors as follows:

\[
sim(V_{\text{الجامعة}}), V_{\text{الكلية}}) = \cos(V_{\text{الجامعة}}), V_{\text{الكلية}}) = 0.13
\]

\[
sim(V_{\text{الجامعة}}), V_{\text{ال مساء}}) = \cos(V_{\text{الجامعة}}), V_{\text{ال مساء}}) = 0.72
\]

That means that, the words "الكلية" (faculty) and "الجامعة" (university) are semantically closer than "ال مـاذ" (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.](https://sites.google.com/site/mohazahran/data)

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:

\[
\begin{align*}
V_1 &= \sum_{k=1}^i v_k \\
V_2 &= \sum_{k=1}^j v'_k
\end{align*}
\]
IDF concept is used to produce a composite weight of tag in the sentence. For example, let \( S_1 \) and \( S_2 \) be two sentences:

\( S_1 = "\text{Joseph went to college}" \)

\( S_2 = "\text{Joseph goes quickly to university}" \)

The similarity between \( S_1 \) and \( S_2 \) is obtained as follows:

**Step 1: Sum of the word vectors**

\[
V_1 = V(\text{كلية}) + V(\text{جامعة}) + V(\text{ يوسف})
\]

\[
V_2 = V(\text{كلية}) + V(\text{جامعة}) + V(\text{ يوسف})
\]

**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 (Salton and Buckley, 1988) and the Part-Of-Speech 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 documents (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: \( idf(w) = \log\left(\frac{S}{WS}\right) \), 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:

\[
\begin{align*}
V_1 &= \sum_{k=1}^{i} idf(w_k) \times v_k \\
V_2 &= \sum_{k=1}^{j} idf(w'_k) \times v'_k
\end{align*}
\]

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:

<table>
<thead>
<tr>
<th>Word</th>
<th>ضروت</th>
<th>اسم</th>
<th>جامع</th>
<th>مضرون</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>0.27</td>
<td>0.37</td>
<td>0.31</td>
<td>0.29</td>
</tr>
</tbody>
</table>

**Step 1: Sum of vectors with IDF weights**

\[
V_1 = V(\text{كلية}) + V(\text{جامعة}) + V(\text{ يوسف}) * 0.31 + V(\text{ يوسف}) * 0.37 + V(\text{ يوسف}) + 0.27
\]

\[
V_2 = V(\text{كلية}) + V(\text{جامعة}) + V(\text{ يوسف}) + 0.34 + V(\text{ يوسف}) + 0.22 + V(\text{ يوسف}) + 0.29 + V(\text{ يوسف}) + 0.37
\]

**Step 2: Calculate the similarity**

The cosine similarity is applied to compute 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, \( \text{verb} = 0.4, \text{noun} = 0.5, \text{adjective} = 0.3, \text{preposition} = 0.1, \text{etc.} \)

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.

\[
\begin{align*}
P_{\text{tag}}(S_1) &= \{ \text{Pos}_{w_1}, \text{Pos}_{w_2}, ..., \text{Pos}_{w_i} \} \\
P_{\text{tag}}(S_2) &= \{ \text{Pos}_{w'_1}, \text{Pos}_{w'_2}, ..., \text{Pos}_{w'_j} \}
\end{align*}
\]

The function \( P_{\text{tag}}(S_1) \) returns for each word \( w_k \) in \( S_i \) its estimated part of speech \( \text{Pos}_{w_k} \).

**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 SemEval2017 (Task 1)\(^3\) to fix the POS weights.

\[
\begin{align*}
V_1 &= \sum_{k=1}^{i} \text{Pos}_{\text{weight}}(\text{Pos}_{w_k}) \times v_k \\
V_2 &= \sum_{k=1}^{j} \text{Pos}_{\text{weight}}(\text{Pos}_{w'_k}) \times v'_k
\end{align*}
\]

where \( \text{Pos}_{\text{weight}}(\text{Pos}_{w_k}) \) 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: \( \cos(S_1, S_2) = \cos(V_1, V_2) \).

\(^2\)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:

<table>
<thead>
<tr>
<th></th>
<th>verb</th>
<th>noun</th>
<th>noun_prop</th>
<th>adj</th>
<th>prep</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>0.7</td>
<td>0.3</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

**Step 1: Pos tagging**
The function $Pos_{tag}(S_i)$ is applied to each sentence.

$$ Pos_{tag}(S_1) = verb \text{ noun} \text{ noun}_{prop} \text{ adj} \text{ prep } $$

$$ Pos_{tag}(S_2) = noun_{prop} \text{ verb} \text{ adj noun } $$

**Step 2: Sum of vectors with POS weighting**

$$ V_1 = V(\text{الكلية}) \ast 0.5 + V(\text{ذهب}) \ast 0.4 $$

$$ V_2 = V(\text{لاجع}) \ast 0.5 + V(\text{مسرة}) \ast 0.3 + V(\text{مضري}) \ast 0.4 + V(\text{ويس}) \ast 0.7 $$

**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:

$$ V_1 = \sum_{k=1}^{n} idf(w_k) \ast Pos_{weight}(Pos_{w_k}) \ast v_k $$

$$ V_2 = \sum_{k=1}^{n} idf(w_k) \ast Pos_{weight}(Pos_{w_k}) \ast v_k $$

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 ” to ” and ” to ”.
3. Replace final ” followed by ” with ”.
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.

We calculate the Pearson correlation between our assigned semantic similarity scores and human judgements. The results are presented in Table 1.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic method (run 1)</td>
<td>0.5957</td>
</tr>
<tr>
<td>IDF-weighting method (run 2)</td>
<td>0.7309</td>
</tr>
<tr>
<td>POS tagging method (run 3)</td>
<td>0.7463</td>
</tr>
<tr>
<td>Mixed method</td>
<td>0.7667</td>
</tr>
</tbody>
</table>

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 Mathematiques Laghouat, Algeria and LIG Laboratoire d’Informatique de Grenoble (GETALP Team), France.
References


