

# Statistical and Semantic Features to Measure Sentence Similarity in Portuguese

Anderson Pinheiro, Rafael Ferreira, Máverick André  
D. Ferreira, Vitor B. Rolim  
Universidade Federal Rural de Pernambuco  
Departamento de Estatística e Informática  
Recife, PE, Brazil  
E-mail: {anderson.pinheiro27, rafaelfmello, amaverick70,  
victor.b.rolim}@gmail.com

João Vitor S. Tenório  
Universidade Federal de Minas Gerais  
Departamento de Ciência da Computação  
Belo Horizonte, MG, Brazil  
E-mail: joaovitorbcc@gmail.com

**Abstract**— A sentence similarity measure is an important field for different applications of text mining. In recent literature, it is possible to find several similarity measures between sentences in English; however, it lacks measures for Portuguese. In addition, one of the main issues to assess sentence similarity is to identify word meaning. In this context, this work aims to present a new approach to measure the similarity between sentences written in Portuguese using statistical and deep learning features to overcome the meaning problems. The results showed that our method obtained better results when compared to the measures proposed in ASSIN 2016 competition.

**Keywords**—Sentence Similarity; Natural Language Processing; Text Mining.

## I. INTRODUCTION

Similarity between sentences becomes important in several applications of text mining, such as information retrieval, text summarization, information extraction and text clustering. For example, in information retrieval, the similarity measure is used to evaluate the relevance of each document for a specific query.

The variability of natural language expression makes it difficult to determine semantic similarity between sentences [1]. One of the main problem to be addressed is the meaning problem; it happens when sentences represent different meanings, even if they have the same words; or otherwise, where sentences with the same meaning, but built with different words [2].

Semantic Similarity has been addressed in several works [3][4][5]. Especially since 2012, when the Semantic Evaluation (SemEval)<sup>1</sup> conference proposed the sentence similarity task, the number of similarity measures for sentences written in English increase [6]. However, the literature lacks efficient measures for other languages.

Recently this task was proposed for Portuguese sentences in the Workshop de *Avaliação Semântica e Inferência Textual* (ASSIN), held during the PROPOR 2016 conference<sup>2</sup>. The main task for ASSIN competition was the semantic textual similarity (STS) where the competitors proposed systems to

determine a semantic similarity value between two sentences varying from 1 to 5.

This paper presents a new measure to calculate the semantic similarity between sentences written in Portuguese. It overcome the meaning problem by applying a deep learning approach to measure similarity among words. The main idea is to combine four different features based on TF-IDF, Direct Matching, Word2Vector and the size of sentences to extract the final similarity.

In addition, this measure was applied to a Recognizing Textual Entailment (RTE) task. RTE is one of the recent challenges of Natural Language Processing (NLP). Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing “Text” (T) and the entailed “Hypothesis” (H). T entails H if the meaning of H can be inferred from the meaning of T [7]. The proposed approach uses the features extracted combined with traditional classifiers to categorize each pair of sentences as Entailment, Paraphrase or Neutral in the RTE task.

The measure was evaluated using the ASSIN database<sup>3</sup>, which has 10,000 pairs of sentences written in Brazilian Portuguese and European Portuguese. The proposed approach outperforms all related methods for both STS and RTE.

## II. RELATED WORK

There are many measures to assess the similarity for sentences written in English. Recently, the number of works proposing measure for other language has increased. For example, in SemEval 2017<sup>4</sup> the STS task will assess the ability of systems to determine the degree of semantic similarity between monolingual and cross-lingual sentences in Arabic, English and Spanish.

The ASSIN task proposed in PROPOR 2016 conference increase the number of papers related to the Portuguese sentence similarity.

Freire, Pinheiro and Feitosa [3] presented a proposal of a framework called FlexSTS, which defines several components to be selected for the development of STS systems, aggregating

<sup>1</sup> <https://www.cs.york.ac.uk/semeval-2012/>

<sup>2</sup> <http://propor2016.di.fc.ul.pt>

<sup>3</sup> <http://nile.icmc.usp.br/assin/>

<sup>4</sup> <http://alt.qcri.org/semeval2017/task1/>

models and similarity measures, toolkits and state of the art algorithms. One drawback of this work was the adoption of WordNet in English, the framework translated words from Portuguese to English in order to use it.

Hartmann [4] reached first place in the ASSIN competition. He used an approach combining classical feature of the bag-of-words, the TF-IDF (Term Frequency-Inverse Document Frequency); and an emerging feature captured through word embeddings. The TF-IDF is used to relate texts which share words. Word embeddings are known by capture the syntax and semantics of a word. The sum of embedding vectors can model the meaning of a sentence. Using both features, the method is able to capture the words shared between sentences and their semantics. Alves, Oliveira and Rodrigues [5] presented two distinct approaches to the ASSIN joint assessment task: a first approach, called Recycling, based exclusively on heuristics under semantic networks for the Portuguese language; and a second approach, dubbed ASAPP, based on supervised automated learning.

Zhao, Zhu and Lan [6] achieved the first ones placed in SemEval 2014 for the task of sentence similarity for English language. The authors extracted several features from sentences to obtain similarity, such as: sentence size, surface similarity (cosine distance), semantic similarity, ngrams based on reference corpus, among others.

Rychalska et al. [8] presented a method of similarity detection that combines recursive autoencoders with a WordNet award-penalty system that accounts for semantic relatedness, and an SVM classifier, which produces the final score from similarity matrices. Barbosa et al. [9] evaluated methods based on semantic word vectors, following two distinct directions: 1) to make use of low-dimensional, compact, feature sets, and 2) deep learning-based strategies dealing with high-dimensional feature vectors.

Pakray, Bandyopadhyay, and Gelbukh [7] present a method based on the decomposition of sentences into three modules, a preprocessing module, a lexical similarity module and a syntactic similarity module. The authors use several features for classification, such as: WordNet based uni-gram match, bi-gram match, longest common sub-sequence, skip-gram, subject-subject comparison, subject-verb comparison, among others. Barreiro [10] studied the paraphrasing of Portuguese phrases based on supporting verbs and analysed the impact of the realization of these paraphrases in the automatic translation of sentences into English. Tsuchida and Ishikawa [11] proposed an RTE system that uses machine learning methods with features based on lexical and predicate argument structure level information.

The proposed measure differs from previous work in two aspects:

- It combines four different features to measure sentence similarity for Portuguese;
- It uses Word2vec and a matrix-based method to deal with the meaning problem;

In addition, it was used a regression system to obtain a similarity value by combining the features for the STS task and different classifiers to RTE task.

### III. BACKGROUND

As mentioned before, the proposed measure extracts four features from sentence pairs to calculate the similarity between sentences. Thus, it is important to highlight TF-IDF and Word2Vec concepts. Besides, the measure benefits from a matrix based method to evaluate similarities among words in sentences [12]. The rest of this sections introduces them.

#### A. TF-IDF

The TF-IDF scheme is a classical approach of NLP. According to Salton [13] TF-IDF is a statistical measure intended to measure the degree of importance of a word to a set of documents (in this work sentences). TF-IDF combines the frequency of terms (TF) and the relevance of the term to a collection (IDF). In this way, the TF-IDF scheme for similarity between sentences is calculated as follows:

$$TF = \left( \frac{\text{number of times a term appears in a given sentence}}{\text{total number of terms present in the sentence}} \right) \quad (1)$$

$$IDF = 1 + \log_e \left( \frac{\text{total number of sentences}}{\text{number of sentences that have a certain term}} \right) \quad (2)$$

$$TF-IDF = TF * IDF \quad (3)$$

At the end is obtained a matrix with *sentences x words* and the value TF-IDF of each word for each sentence. The similarity between the sentences is calculated by the cosine distance between the TF-IDF vectors of the sentence pairs.

#### B. Word2Vec

The Word2vec is an unsupervised model to generate a vector representation of each word in a word set [14]. The main goal of this representations is measure semantic similarities among words.

Word2vec predicts the neighbors of a word using a neural network algorithms. There are two possible types of prediction: Skip-gram-distributed or Continuous Bag-Of-Words (CBOW) [15]. The CBOW predicts the current word based on the words that are around it. On the other hand, Skip-gram model predicts the neighboring words given the current word. The word vector corresponds to the weights between the input and the first hidden layer in the feed-forward network used. The size of the final vector space is an input parameter. Word2vec uses skip-gram because it produces more accurate results for large data sets.

After the training step, Word2vec simplifies context of a word into a K-dimensional vector space. Therefore, this representation could be used to obtain the similarity between words. The similarity value between words is obtained by calculating the cosine distance between the vectors of each word.

### C. Matrix-Based Method

Ferreira et al. [12] present a three layers sentence representation to calculate the similarity between a pair of sentences written in English. The layers are: (i) shallow layer, which compose the lexical analysis, stopwords and named entity recognizer (NER); (ii) syntactic layer, which constitutes the syntactic analysis, NER, and the coreference relations; and (iii) semantic layer, which mainly describes the semantic paper annotation.

This approach used a matrix-based method to calculate the similarity between sentences. The similarity is compounded by the similarity between words. It follows the steps to calculate this measure.

The first step calculates the similarity among the words of two sentences. Let  $A = \{a_1, a_2, \dots, a_n\}$  and  $B = \{b_1, b_2, \dots, b_m\}$  be two sentences, where  $a_i$  is a word from sentence A,  $b_j$  is a word from sentence B,  $n$  is the number of words from sentence A and  $m$  is the number of words from sentence B. Then it is calculated the similarity value between each word of sentence A with each word of sentence B. For example, be two sentences A and B, each sentence with six words each. The similarities of all words of sentence A with all words of sentence B are calculated, as shown in Figure 1. The highest similarity was obtained, 1.0, was between the words  $a_4$  and  $b_6$ .

The second step is to remove the words that had the highest similarity in the previous step. In the example above were removed the words  $a_4$  and  $b_6$ , Figure 2.

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$b_1$	0.3	0.2	0.56	0.88	0.25	0.13
$b_2$	0.12	0.5	0.31	0.22	0.87	0.65
$b_3$	0.56	0.23	0.5	0.28	0.6	0.63
$b_4$	0.7	0.62	0.6	0.38	0.12	0.1
$b_5$	0.84	0.21	0.54	0.78	0.29	0.56
$b_6$	0.4	0.35	0.47	<b>1.0</b>	0.23	0.33

Fig. 1. Example of similarities between words.

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$b_1$	0.3	0.2	0.56	0.88	0.25	0.13
$b_2$	0.12	0.5	0.31	0.22	0.87	0.65
$b_3$	0.56	0.23	0.5	0.28	0.6	0.63
$b_4$	0.7	0.62	0.6	0.38	0.12	0.1
$b_5$	0.84	0.21	0.54	0.78	0.29	0.56
$b_6$	<del>0.4</del>	<del>0.35</del>	<del>0.47</del>	<del><b>1.0</b></del>	<del>0.23</del>	<del>0.33</del>

Fig. 2. The words  $a_4$  and  $b_6$  are removed of the matrix.

The steps 1 and 2 are repeated until they have no more words to calculate the similarity. The last step involves averaging between the highest values of similarities obtained between the sentences. The average is calculated as follows:

$$\text{SentenceSim}(A, B) = \frac{\sum_{i=1}^n \text{MaxSim}(a_i, b_n)}{n} \quad (4).$$

The similarity value between sentence A and B will be the average of the highest similarities obtained between each word of sentence A and each word of sentence B.

### IV. PROPOSED METHOD

The proposed approach extracts four features from sentence pairs and different classifiers to compute the similarity and identify entailment. Each feature and classifiers used are presented in the following sections.

#### A. Extraction of features

1) *TF-IDF*: This feature benefits from the cosine distance between the each sentence TF-IDF vectors to calculate de similarity. Before extract TF-IDF values, two preprocesses were adopted: (i) stemming, in order to reduce the sparsity of the data [16]; and a word expansion, it expand the synonyms for words of content that have up to 2 synonyms in TEP (Thesaurus for Brazilian Portuguese) [17]. These pre-processing techniques were adopted based on the tests results from previous work [4].

2) *Word2Vec*: The second feature was obtained using an matrix-based method, shown in section III-C, combined with Word2vec word similarity measure. The Word2vec model was built using the original implementation<sup>5</sup> on the basis of wikipedia<sup>6</sup> and texts of news obtained from website G1<sup>7</sup>. Standard preprocessing techniques were performed (for example, lowercase, punctuation removal). To build the model the following Word2vec parameters were used:

- dimension: 250;
- Window: 10;
- Minimum word frequency: 5;
- Number of iterations: 10.

In addition, the sentences passed through a stopwords removal and lemmatization process before the similarity calculation.

3) *Binary Matrix Method*: The previous method usually reaches high similarities values, even for sentences with low similarity value. Therefore, it was proposed a binary matrix method to overcome this problem. It also uses the matrix-based approach, the difference is that the similarity values between the words are obtained as follows:

$$\text{sim}(a, b) = \begin{cases} 1, & \text{if the words are equals} \\ 0, & \text{if the words are different} \end{cases} \quad (5).$$

At the end is obtained the average of the similarities between the words to obtain the similarity between the sentences. For this method the stopwords removal and stemming techniques were used.

4) *Sentence Size*: The last feature, also used by Zhao, Zhu and Lan [6] and Bjerva et al. [18], was the size of the sentences. To obtain a value that represents the size of the

<sup>5</sup> <http://code.google.com/p/word2vec>

<sup>6</sup> <https://dumps.wikimedia.org/ptwiki/20160920/>

<sup>7</sup> <http://g1.com.br/>

sentences, the number of words of the lowest sentence is divided by the number of words of the highest sentence. For this method the stopwords were removed.

### B. Classifiers

A regression algorithm was used to combine the features in order to quantify the final similarity. In addition, different classifiers were adopted to categorize entailment sentences.

1) *Regression*: The regression consists in the execution of a statistical analysis in order to verify the existence of a functional relation between a dependent variable with one or more independent variables. It was used linear regression algorithm because it obtained the best results in comparison with other types of regression [4]. The algorithm receives the extracted features as input with similarity proposed on training set and the output is a function to combine these features generating the final similarity on test set.

2) *Support Vector Machine (SVM)*: Support Vector Machines (SVMs) are a technique based on Statistical Learning Theory [19]. According to Burgues [20], to perform classification/recognition of pattern the SVM constructs hyperplanes in a multidimensional space aiming to separate cases of different classes. When the SVM separates the vectors of the classes without error and with maximum distance to the nearest vectors is considered as optimal separation [21]. However, some problems may not be separable linearly, in these cases the SVM uses kernels functions that, in turn, allow the mapping of the data to a larger dimensional space, in order to enable the linear separation. In this work, the kernel rbf function was used.

3) *Naive Bayes*: The Naive Bayes classifier is based on the Bayes theorem and its main feature is to assume that all the attributes of the examples are independent of each other, given the context of the class [22]. This is Bayes' "naive" assumption. While this assumption is clearly false in most real-world tasks, Naive Bayes has several reports in the literature about its competitiveness toward other classifiers.

4) *Neural Network*: Artificial Neural Networks are computational techniques that present a mathematical model inspired by the neural structure of intelligent organisms and that acquire knowledge through experience. This model is composed of a set of neurons, or nodes, that are interconnected with each other, forming a network. Each neuron receives inputs with an associated weight. From the inputs and their respective weights, a weighted summation is performed at the nucleus of the neuron and based on an activation threshold it is checked whether or not the input will propagate to neurons of the adjacent layers of the current layer. In this work we used a Multilayer Perceptron (MLP) that consists of a classic neural network model [23]. In this work 4 neurons are used in the input layer, where for each node a feature of the sentence pair to be classified is assigned. The output layer contains three neurons, where each neuron corresponds to a class: Entailment, Paraphrase or Neutral.

## V. EXPERIMENTS

This section describes the database and evaluation measures for the STS task and for the RTE task. Then, the proposed measure results are presented.

### A. Database

The ASSIN database (ASSIN, 2016) has 10,000 (ten thousand) sentence pairs, 5,000 (five thousand) in Brazilian Portuguese and 5,000 (five thousand) in European Portuguese. For each 5,000 sentences, 3,000 (three thousand) are for training and 2,000 (two thousand) are for testing. Each pair of sentences has a semantic similarity value, ranging from 1 to 5, and a class to which it belongs (Entailment, Paraphrase or Neutral).

### B. Evaluation Measures

The measures used in this work were the measures adopted by the ASSIN competition. We used the same measures to compare our measure with the measures proposed in the competition. For STS task evaluation were used the Pearson Correlation coefficient (PC) and the Mean Square Error (MSE) to measure the degree of correlation between the similarities automatically obtained by competitors' systems and the similarities present in the database. The PC ranges from -1 to 1. The signal indicates positive or negative direction of the relationship and the value suggests the strength of the relationship between the variables. A zero value correlation indicates that there is no linear relationship between the variables. The best results has PC closer to 1; it means a greater degree of statistical dependence between the variables [24]. The PC is calculated as follows:

$$PC(x,y) = \frac{cov(x,y)}{\sqrt{var(x) * var(y)}} \quad (6),$$

where  $x$  is the obtained similarity value and  $y$  is the desired similarity value.

The MSE is defined as the sum of the differences between the estimated value and the actual value of the data, weighted by the number of terms. MSE is calculated as follows:

$$MSE(Y,Y') = \sum_i^n \frac{(y_i - y'_i)^2}{n} \quad (7),$$

where  $Y$  is the set of estimated values and  $Y'$  is the set of real values and  $n$  is the number of terms. Lower values of MSE means a smaller values of error. Thus, the best values of MSE are close to 0.

The evaluation for RTE task adopts Accuracy and F-measure measures. F-measure uses Precision and Recall measurements. These measures are calculated as follows:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (8),$$

$$precision = \frac{TP}{(TP+FP)} \quad (9),$$

$$recall = \frac{TP}{(TP+FN)} \quad (10),$$

$$f\text{-measure} = 2 * \frac{precision * recall}{precision + recall} \quad (11),$$

where, to calculate the precision and recall is used a confusion matrix (Figure 4) known to tabulate the results obtained as follows: True positive (TP) - number of positive elements classified as positive; True negative (TN) - number of positive elements classified as false; False positive (FP) - number of false elements classified as positive and; False negative (FN) - number of false elements classified as false. In view of this, Zhang and Zhang [25] define f-measure as a weighted measure of precision and recall (11).

		True class	
		positive	negative
Predicted class	positive	<b>TP</b> True Positive	<b>FP</b> False Positive
	negative	<b>FN</b> False Negative	<b>TN</b> True Negative

Fig. 3. Confusion matrix.

### C. Results

The experiments were conducted to evaluate semantic similarity using 2 linear regression systems, one for the Brazilian Portuguese training data set (PTBR) and another for the European Portuguese training data set (PTPT) of ASSIN database. The systems were trained using each feature individually extracted and the combination of the four features. The systems were evaluated using the ASSIN test data set with 2,000 (two thousand) sentence pairs for PTBR and 2,000 (two thousand) sentence pairs for PTPT.

Table I shows the results obtained adopting the acronyms: Word2Vec Matrix-based Method (WMM), Binary Matrix Method (BMM) and Sentence Size (SS). The intervals from 0 to 1 were converted from 1 to 5 and then were calculated the PC and MSE of the similarities obtained with the similarities of the ASSIN database.

TABLE I. PC AND MSE OBTAINED USING LINEAR REGRESSION FOR STS TASK.

Feature	PTBR		PTPT	
	PC	MSE	PC	MSE
TF-IDF	0.67	0.62	0.65	0.64
WMM	0.64	0.88	0.68	1.5
BMM	0.61	1.63	0.64	1.17
SS	0.10	2.18	-0.04	3.1
TF-IDF + WMM	0.70	0.38	0.70	0.74
TF-IDF + BMM	0.67	0.96	0.68	0.74
WMM + BMM	0.64	0.48	0.67	0.62
TF-IDF + WMM + SS	0.66	1.21	0.66	1.21
TF-IDF + BMM + SS	0.67	0.43	0.66	0.67
WMM + BMM + SS	0.65	0.40	0.67	0.92
TF-IDF + WMM + BMM	0.70	0.38	0.70	<b>0.57</b>
All features	<b>0.71</b>	<b>0.37</b>	<b>0.71</b>	0.63

The results on table I shows that the combination of all features achieve better result for Brazilian Portuguese and

better PC for PTPT. In addition, the combination of TF-IDF + WMM + BMM reaches better value for MSE for European Portuguese.

Table II compares the best result obtained with the proposed measure with the results obtained by the ASSIN 2016 competition teams. For the PTBR it was used all the features in the linear regression system, and for the PTPT the combination adopted was the TF-IDF, Matrix Method and Binary Matrix Method because they obtained the smallest error. The two linear regression systems (PTBR and PTPT) return a similarity value by combining these features. The total column compares the similarity values obtained for the entire test database (4,000 sentences) contained in ASSIN database.

TABLE II. COMPARISON OF PROPOSED METHOD WITH ASSIN COMPETITION TEAMS FOR STS TASK.

Team/Method	PTBR		PTPT		TOTAL	
	PC	MSE	PC	MSE	PC	MSE
Proposed Measure	<b>0.71</b>	<b>0.37</b>	0.70	<b>0.57</b>	<b>0.70</b>	<b>0.47</b>
Solo Queue	0.70	0.38	0.70	0.66	0.68	0.52
Reciclagem	0.59	1.31	0.54	1.10	0.54	1.23
Blue Man G.	0.65	0.44	0.64	0.72	0.63	0.59
ASAPP	0.65	0.44	0.68	0.70	0.65	0.57
LEC-UNIFOR	0.62	0.47	0.64	0.72	0.62	0.59
L2F/INESC-ID			<b>0.73</b>	0.61		

Table II shows that the proposed measure outperforms all related work. It achieve the best results for PTBR and in the total result (PTBR + PTPT), for PTPT our method obtained PC below the L2F/INESC-ID team, but obtained the best MSE. It is important to notice that L2F/INESC-ID team was ranked first in the PTPT competition, however this team did not present a measure for PTBR. Thus, the proposed measure presented the best result in total for Brazilian Portuguese and European Portuguese.

For the RTE task the experiments were conducted using the measures F-measure (F1) and Accuracy (percentage of correctly classified instances) with the classifiers shown in Subsection IV-C. The results are shown in Table III.

TABLE III. RESULTS OF CLASSIFIERS USED FOR RTE TASK.

	PTBR		PTPT	
	Accuracy	F1	Accuracy	F1
<i>SVM</i>	84.85%	0.817	82.35%	0.805
<i>Naive Bayes</i>	83.90%	0.840	82.25%	0.823
<i>MLP</i>	<b>85.35%</b>	<b>0.811</b>	<b>82.75%</b>	<b>0.819</b>

As can be seen from Table III, the MLP obtained the best accuracy with 85.35% and 0.811 of F1 for PTBR and 82.75% accuracy and 0.819 of F1 for PTPT. Table IV compares the result obtained with the MLP with the results of the competition teams.

As presented on Table IV, the proposed measure obtained the best results, losing only in the accuracy for the L2F/INESC-ID team. The results are expressive for f-measure it reaches 55.96%, 34.26% and 42.93% better results in relation

to the competitors for PTBR, PTPT and TOTAL respectively. For Accuracy the proposed measure achieves values 4.53% and 3.46% higher than competitors for PTBR and TOTAL respectively.

TABLE IV. COMPARISON OF PROPOSED METHOD WITH ASSIN COMPETITION TEAMS FOR RTE TASK.

Team/Method	PTBR		PTPT		TOTAL	
	Acc.	F1	Acc.	F1	Acc.	F1
Proposed Measure	<b>85,35%</b>	<b>0,811</b>	82,75%	<b>0,819</b>	<b>83,05%</b>	<b>0,829</b>
Reciclagem	79,05%	0,39	73,10%	0,43	75,58%	0,38
Blue Man G.	81,65%	0,52	77,60%	0,61	79,62%	0,58
ASAPP	81,65%	0,47	78,90%	0,58	80,27%	0,54
L2F/INESCID			<b>83,85%</b>	0,7		

## VI. FINAL CONSIDERATIONS

The similarity between sentences becomes important in several natural language applications such as text summarization, information extraction and text grouping. In this work a new measure to calculate the similarity between sentences written in Portuguese was presented.

This measure uses four features extracted from sentences pairs. For the STS task a linear regression system was used to obtain a similarity value combining the extracted features. For the task of RTE, 3 classifiers were analysed, where the MLP obtained 85.35% accuracy and 0.811 F1, the best result in comparison with the others. For both tasks of the ASSIN competition our method, in general, achieved the best result when compared to the systems presented on the competition.

As future work we intend to train Word2vec with texts in Portuguese from Portugal to obtain better results with sentences written in European Portuguese. We also want to validate a metric with plagiarism detection in an educational forum, where each new post in the forum will be analysed using our metric in relation to other forum posts. If this post is very similar to those in the forum, our system will send a message to the student requesting to modify the text of your post. In this way this system can help reduce or even eliminate the copy of ideas, concepts or works of other authors and help in the construction of the ethical understanding of each student.

## REFERENCES

- [1] P. Achananuparp, X. Hu, and S. Xiajiong, "The evaluation of sentence similarity measures," *Lecture Notes in Computer Science*, vol. 5182, pp. 305-316, 2008.
- [2] B. Choudhary and P. Bhattacharyya, "Text clustering using semantics," in *Proceedings of World Wide Web Conference 2002 (WWW'02)*, 2002.
- [3] J. Freire, V. Pinheiro and D. Feitosa "LEC\_UNIFOR no ASSIN: FlexSTS - Um Framework para Similaridade Semântica Textual," *Workshop de Avaliação de Similaridade Semântica e Inferência Textual*, 2016.
- [4] N.S. Hartmann, "Solo Queue at ASSIN: combinando abordagens tradicionais e emergentes," In: *Workshop de Avaliação de Similaridade Semântica e Inferência Textual*, 2016.

- [5] A.O. Alves, H.G. Oliveira and R. Rodrigues, "ASAPP: Alinhamento Semântico Automático de Palavras aplicado ao Português," In: *Workshop de Avaliação de Similaridade Semântica e Inferência Textual*, 2016.
- [6] J. Zhao, T.T. Zhu and M. Lan, "ECNU: One Stone Two Birds: Ensemble of Heterogenous Measures for Semantic Relatedness and Textual Entailment," In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, August 2014, pp. 271-277.
- [7] P. Pakray, S. Bandyopadhyay and A. Gelbukh, "Textual entailment using lexical and syntactic similarity," *Internacional Journal of Artificial Intelligence and Applications*, vol. 2, no. 1, 2011, pp. 43-58.
- [8] B. Rychalska, K. Pakulska, K. Chodorowska, W. Walczak and P. Andrzejewicz, "Necessity for diversity; combining recursive autoencoders," in: *Proceedings of 10th workshop on semantic evaluation (SemEval 2016)*, June 2016, pp. 602-608.
- [9] L. Barbosa P. Cavalin, M. Kormaksson and V. Guimarães, "Blue Man Group at ASSIN: Using Distributed Representations for Semantic Similarity and Entailment Recognition," in *Workshop de Avaliação de Similaridade Semântica e Inferência Textual*, 2016.
- [10] A. Barreiro, "ParaMT: A Paraphraser for Machine Translation," *8th International Conference, PROPOR*, pp. 202-211, 2008.
- [11] M. Tsuchida and K. Ishikawa, "A method for recognizing textual entailment using lexical-level and sentence structure-level features," in *Proceedings of the Text Analysis Conference*, 2011.
- [12] R. Ferreira, R.D. Lins, S.J. Simske, F. Freitas and M. Riss, "Assessing sentence similarity through lexical, syntactic and semantic analysis," In: *Computer Speech & Language*, vol. 39, September 2016, pp. 1-28.
- [13] G. Salton and C.S. Yang, "On the specification of term values in automatic indexing," In: *Journal of Documentation*, vol. 29, no. 4, pp. 351-372, 1973.
- [14] D.R.G.H.R. Williams and G. Hinton, "Learning representations by back-propagating errors," *Nature*, pp. 523-533, 1986.
- [15] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.
- [16] G. Pedrosa, M. Pita, P. Bicalho, A. Lacerda and G.L. Pappa, "Topic Modeling for Short Texts with Co-occurrence Frequency-based Expansion," in *5th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct. 2016, pp. 277-282.
- [17] E. Maziero and T. Pardo, "Interface de Acesso ao TeP 2.0 - Thesaurus para o português do Brasil," Technical report, University of São Paulo, 2008, unpublished.
- [18] J. Bjerva, J. Bos, R.V.D. Goot and M. Nissim, "The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity," in *2014 International Workshop on Semantic Evaluation*, 2014, pp. 642-646.
- [19] V.N. Vapnik, "The nature of Statistical learning theory," *Springer-verlag*, New York, 1995.
- [20] C.J.C. Burgues, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, no. 2, pp. 1-43, 1998.
- [21] V.N. Vapnik, "The nature of Statistical learning theory," *2nd edition Springer-verlag*, New York, 1999.
- [22] I. Rish, "An empirical study of the naive bayes classifier," in *Workshop on empirical methods in artificial intelligence*, 2001.
- [23] S.B. Wankhede, "Analytical study of neural network techniques: SOM, MLP and Classifier-A survey," in *Journal of Computer Engineering*, vol. 16, no. 3, 2014, pp. 86-92.
- [24] B. Dalson and A. José, "Desvendando os Mistérios do Coeficiente de Correlação de Pearson (r)," in *Revista Política Hoje*, vol. 18, no. 1, pp.115-148, 2009.
- [25] E. Zhan and Y. Zhang, "F-measure," in *Encyclopedia of DataBase Systems*, Springer US, pp. 1147-1147, 2009.