Blue Man Group at ASSIN: Using Distributed Representations for Semantic Similarity and Entailment Recognition

Blue Man Group no ASSIN: Usando Representações Distribuídas para Similaridade Semântica e Inferência Textual

Luciano Barbosa
IBM Research
lucianoa@br.ibm.com

Paulo Cavalin
IBM Research
pcavalin@br.ibm.com

Victor Guimarães
IBM Research
victorl@br.ibm.com

Matthias Kormaksson
IBM Research
matkorm@br.ibm.com

Abstract

In this paper, we present the methodology and the results obtained by our team, dubbed Blue Man Group, in the ASSIN (from the Portuguese *Avaliação de Similaridade Semântica e Inferência Textual*) competition, held at PROPOR 2016[1].

Our team's strategy consisted of evaluating 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. Evaluation results demonstrated that the first strategy was more promising, so that the results from the second strategy have been discarded.

As a result, by considering the best run of each of the six teams, we have been able to achieve the best accuracy and F1 values in entailment recognition, in the Brazilian Portuguese set, and the best F1 score overall. In the semantic similarity task, our team was ranked second in the Brazilian Portuguese set, and third considering both sets.

Keywords

Semantic Similarity, Entailment Recognition, Deep Learning, Word Vectors

Resumo

Neste artigo apresentamos a metodologia e os resultados obtidos pela equipe Blue Man Group, na competição de Avaliação de Similaridade Semântica e Inferência Textual do PROPOR 2016.

A estratégia da equipe consistiu em avaliar métodos baseados no uso de vetores semânticos de palavras, com duas frentes básicas: 1) uso de vetores de características de pequena dimensão, e 2) estratégias de deep learning para vectores de características de grandes dimensões. Os resultados nas bases de avaliação demonstraram que a primeira frente seria mais promissora, e os resultados submetidos para a competição da segunda frente foram descartados.

Com isso, considerando o melhor resultado de cada uma das seis equipes, conseguimos atingir os melhores resultados de acurácia e medida F1 na tarefa de inferência textual, na base de português brasileiro, e o melhor resultado geral de F1 considerando ambas as bases. Na tarefa de similaridade semântica, a equipe atingiu o segundo lugar na base de português brasileiro, e terceiro lugar considerando ambas as bases.

Palavras chave

Similaridade Semântica, Inferência Textual, Deep Learning, Word Vectors

1 Introduction

In this work, we present the methodology and results obtained by our team, dubbed Blue Man group, in the *Avaliação de Similaridade e Inferência Textual* (ASSIN) competition, jointly held with the International Conference on the Computational Processing of Portuguese (PROPOR) 2016.

The ASSIN competition assigned two tasks to participants: evaluation of semantic similarity, and entailment recognition. Given sentences s_1 and s_2, the first task consists of providing a score ranging from 1 to 5, representing the strength of the semantic relationship between s_1 and s_2. The second task involves determining whether s_1 entails s_2 (a sentence s_1 entails another sentence s_2 if, after reading both and knowing that s_1 is true, a person concludes that s_2 must also be true). Given these two tasks, researchers are in-

[1] International Conference on the Computational Processing of the Portuguese Language - http://propor2016.di.fc.ul.pt/

vited to form teams and participate in the competition by developing systems that solve either or both of them, by making use of labeled data provided by the organization of the competition, and submit their results on a blind test data. It is worth mentioning that sets with text in Portuguese from both Brazil and Portugal were available, i.e. PT-BR and PT-PT, and teams could choose to submit results for either or both sets.

Our team (Blue Man Group) focused on word vectors-based approaches to solve both tasks (see details in Section 3). By considering word vectors created with the entire Portuguese Wikipedia, we have followed two distinct directions. In the first, we implement a state-of-the-art feature set, proposed in (Kenter e de Rijke, 2015), to train both support vector regression/classification models and Lasso regression. In the second direction, we exploit deep-learning setups of siamese neural networks. Preliminary evaluations on the training and trial data sets demonstrated that the first direction was more promising, and we have decided to submit the results of that methodology only.

In total, six teams participated in the competition. By considering the best run of each team, our system worked best in the entailment recognition task, ranking first in both accuracy and F1 for the PT-BR set, while ranking second in accuracy and first in F1 overall. In the semantic similarity evaluation, our best results were ranked second in both Pearson correlation and Mean Squared Error (MSE) for the PT-BR set, while ranking second in Pearson and third in MSE overall. For the PT-PT set, the system performed better for entailment recognition, achieving the second best F1 score, while achieving only the 4th place in semantic similarity.

In the remainder of this document we present details on how our system was developed and evaluated.

2 ASSIN Competition

The ASSIN competition, a.k.a. *Avaliação de Similaridade Semântica e Inferência Textual*, consists of an evaluation forum for two NLP-related tasks, i.e., semantic similarity and textual entailment recognition, in which registered participants (or teams) could develop systems and submitted their results on the data provided by the organizing committee. A large dataset containing pairs of sentences, in both Portugal's and Brazil's variants of Portuguese, has been created to allow participants to both develop and evaluate the systems. Participants could submit results to either or both tasks, and also either or both variations of Portuguese. Then, the teams would be ranked by the results of their systems on the evaluation dataset, namely the test set. Both the metrics and the sets, as well as the tasks, are explained in details in the remaining of this section.

The ASSIN dataset, containing a total of 10,000 pairs of sentences, can be divided in the following subsets. The Brazilian training set contains 3,000 labelled pairs of sentences collected from Google News, from Brazilian sources. The Portuguese training set also contains 3,000 labelled pairs of sentences collected from Google News, but from Portuguese sources. And the Brazilian and Portuguese blind test sets, contain 2,000 unlabeled pairs of sentences each, from the same sources. It is worth mentioning that the labels of the test sets have been released to the participants only after they had submitted their results.

For the first task, i.e. semantic similarity, the semantic relatedness is measured in a scale from 1 to 5, where 1 stands for completely different sentences, and 5 sentences that means essentially the same thing. The scales in between are gradual variations of these two concepts. In the light of this, this task consists of building a model which, given the pair of sentences $p(i) = \{s_1(i), s_2(i)\}$, containing sentence $s_1(i)$ and sentence $s_2(i)$, predicts the semantic similarity score $y(i)$. Given the manually-labeled similarity scores $x(i)$, systems are evaluated by means of the Pearson correlation between the set containing all $x(i)$ and $y(i)$, for $i > 0$, and the Mean Squared Error (MSE).

The second task – recognizing textual entailment (RTE) – consists of determining whether the meaning of the hypothesis is entailed from the text (RTE, 2011). That is, suppose s_1 is the text and s_2 is the hypothesis, s_1 entails s_2 if, after reading both and knowing that s_1 is true, a person concluded that s_2 must also be true. Given that the dataset provided by ASSIN also distiguishes bidirectional entailment cases, or paraphrases, the pair of sentences s_1 and s_2 must be classified into one of the following classes: *entailment*, *paraphrase*, and *no relation*. Given the ground-truth labels, systems are measured by means of accuracy and F1 score.

3 Methodology

As already mentioned, the strategy employed by our team consisted in evaluating word vector-based approaches, where the word vectors represent the semantic meaning of words (see Section 3.1). As a result, two distinct directions have been followed. The first, presented in Section 3.2, consists of implementing a state-of-the-art feature set for representing the similarity relatedness of pairs of sentences, and using regression models such as support vector regression (SVR) for semantic similarity evaluation, and support vector machines (SVM) for entailment recognition. And the second, in Section 3.3, exploits deep-learning siamese neural networks, with the goal of learning better representation from raw data, i.e. the word vectors of the pair of sentences.

3.1 Word vectors

Word vectors (or word embeddings) have been successfully used over the past years to learn useful word representations, encoding the semantic meaning of words by means of continuous vectors (Collobert et al., 2011). In other words, even if two words are lexically written in two very distinct ways, if these two words present similar semantic meaning, their corresponding word vectors should be very similar. These vectors make it possible not only to create NLP methods that rely more on the semantic meaning of the words than on their lexical form, but to take advantage of large corpora of text since the learning of word vectors can be done in an unsupervised fashion.

The learning of word vectors is done in the following way. Given a large corpus of text, word vectors are learned by considering the distributional frequency of words. That is, given a word and its preceding and subsequent words in a sentence, a machine learning model such as a neural network can be learned by using the neighbouring words are input, and the central word as output.

In this work, word vectors have been created with the *word2vec* tool[2], using the entire Portuguese Wikipedia as input. This set contains a total of 636,597 lines of texts, with 229,658,430 word occurrences, and a vocabulary of size 540,638. The *word2vec* tool was setup with: skip n-grams model; word vector size equals to 300; maximum skip length between words set to 5; 10 negative samples; hierarchical softmax not used; threshold of occurrence of words set to 10e-5; and 15 training iterations.

[2]http://code.google.com/archive/p/word2vec/

3.2 Strategy 1: Kenter's features

3.2.1 Feature set

The feature set proposed in (Kenter e de Rijke, 2015), consists of extracting a single feature vector, denoted $\bar{x}_i = x_{i1}, \ldots, x_{iK}$, to encode the semantic similarity from the pair of sentences $s_1(i)$ and $s_2(i)$. In this work, we propose the use of such feature set for both tasks in the competition, i.e. semantic similarity evaluation and entailment recognition.

Given the sets of word vectors $\Omega_{i,1}$ and $\Omega_{i,2}$, computed from sentences $s_{i,1}$ and $s_{i,2}$, the feature set is composed of two types of features. 1) semantic networks; and 2) text-level features.

In short, semantic networks consist of building a network considering the distances of pairs of word vectors $(\omega_{1,j}, \omega_{2,k})$ that appear in $s_{i,1}$ and $s_{i,2}$, where $\omega_{1,j} \in \Omega_{i,1}$ and $\omega_{2,j} \in \Omega_{i,2}$. In this case, two types of networks are built. The first, namely Saliency-weighted Semantic Network, combines both similarity and inverse document frequency (IDF) to create the links between the nodes, by considering, for each word vector $\omega_{1,j}$ in $\Omega_{i,1}$, the most similar word vector $\omega_{2,k}$ in $\Omega_{i,2}$, i.e. the word vector $\omega_{2,k}$ with the smallest cosine distance to $\omega_{1,j}$. The links in the weighted network represent the distances between the corresponding word vectors, multiplied by the IDF of the corresponding term from $s_{i,1}$. In this work, the IDF is computed in the same set used to created the set of word vectors, i.e. the Portuguese Wikipedia. The second type of network, referred to as Unweighted Semantic Network, in contrast, does not rely on IDF, and two different unweighted networks are derived from this. One contains the distances of all pairs of terms $(\omega_{1,j}, \omega_{2,k})$. And in the other, only the pairs $(\omega_{1,j}, \omega_{2,k})$ with minimal distances are considered, such as with saliency-weighted semantic networks.

In the end, the information in the networks mentioned in the previous paragraph is used to create histograms, which are concatenated to compose a single feature vector. The boundaries for these histograms have been defined in the following way. For the features calculated from the saliency-weighted semantic network, the values are 0-.15, .15-.4, and .4-∞. For both unweighted semantic networks, the values are -1-.45, .45-.8, and .8-∞.

Besides, the feature set also includes text-level features. These features are defined in two ways: 1) distance between word vectors, where both the cosine and Euclidean distances are computed between the mean word vectors of $s_{i,1}$ and $s_{i,2}$;

and 2) bins of dimensions, where a histogram is computed from the real values presented in the mean word vectors of the pair of sentences. In this case, the boundaries for the histogram have been defined as $-\infty$-.001, .001-.01, .01-.02, and .02-∞.

The resulting feature set is then composed of a 15-position vector, corresponding to: 3 features from histogram of saliency-weighted semantic networks, 2×3 from the histograms from the two unweighted semantic networks, 2 from the distances of the mean word vectors, and 4 from the bins of dimensions.

In addition, it is worth mentioning that these 15 features can be replicated by making use other sets of word vectors. In other words, for each distinct set of word vectors, a new 15-position feature vector can be extracted, and these feature vectors can be combined. In this work, though, we consider a single set of word vectors, i.e. the one described in Section 3.1, for the sake of simplicity.

More details about this feature set, such as how the boundaries of the histograms have been defined, can be found in (Kenter e de Rijke, 2015).

3.2.2 Support Vector Regression and Support Vector Machines

Support vector machines (SVM), and their corresponding method for regression problems, i.e. Support Vector Regression (SVR), have become popular in the past years given the good performance in a high number of tasks (Byun e Lee, 2002). SVM and SVR employ the following idea: input vectors, denoted $x_{i1}, \ldots, x_{iK}$, are non-linearly mapped to a very high-dimension feature space. In this feature space, a linear decision surface is constructed, in order to predict the class value $y_i \in [-1, 1]$, in the case of classification, or the target real value y_i, in the case of regression. Special properties of the decision surface ensures high generalization ability of the learning machine (Cortes e Vapnik, 1995).

For this work, both SVR and SVM have been implemented with the *Scikit Learn* library[3]. For both methods, we used the Gaussian kernel after a few preliminary experiments. And the configuration parameters of both have been setup by means of a grid search with five-fold cross validation on the training set.

[3]http://scikit-learn.org

3.2.3 Lasso

Let y_i denote the response and let $x_{i1}, \ldots, x_{iK}$ denote the K features calculated for each observation i. We considered the following regression model:

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ik} + \sum_{\ell \neq k} \alpha_{\ell k} x_{i\ell} x_{ik} + \varepsilon_i,$$

where ε_i denotes the error associated with observation i. The above model is linear in the features and includes all possible two-way interactions, $x_{i\ell}x_{ik}$, between pairs of features. Let θ denote the set of all parameters $(\beta_k)_k$ and $(\alpha_{\ell k})_{\ell k}$. By correctly specifying a design matrix X (whose columns are the features and corresponding two-way interactions) we may formulate the above regression in a more simple matrix notation:

$$y = X\theta + \varepsilon,$$

where y and are the response and error vectors respectively.

Note that if we were to estimate the above model using the method of least squares we could easily have problems with over-fitting due to the large amount of parameters to be estimated:

$$n_{param} = K + 1 + \frac{(K-1) \cdot K}{2} \sim O(K^2).$$

Lasso regression (Tibshirani, 1996) is designed to tackle this potential problem of over-fitting and falls into a class of models called regularized regression. By applying least squares with an additional L_1-constraint on the parameters, $\|\theta\|_1 = \sum_k |\theta_k| \leq C$, for some $C > 0$, we are able to guard against over-fitting. This method has an advantage in that it serves as a method for variable selection as well, since the L_1-penalty effectively forces some of the parameter estimates to be exactly equal to 0.

3.3 Strategy 2: Siamese Networks

Siamese networks (Chopra, Hadsell e LeCun, 2005) have been widely used in image and text processing to learn a similarity metric from data. For the specific task proposed on ASSIN, we use siamese networks to learn the similarity between two sentences in Portuguese. Essentially, given a pair of sentences, a siamese network projects each sentence in a new representation space using, for instance, convolutional or recurrent networks. The parameters W of each sentence projection are shared. These representations are then given as input to a pre-defined similarity metric such as cosine or euclidean that calculates the similarity between the two representations. During training, the network learns the values of W that minimize a given loss function. In our experiments, we use Mean Squared Error as the loss function. The error is the difference between the true similarity value given in the training data and the predicted one. From this framework, we tried different configurations. For instance, to project the sentences we tried convolutional (CNN) (Collobert et al., 2011) and a type of recurrent networks called long short-term memory network (LSTM) (Hochreiter e Schmidhuber, 1997). We use cosine similarity as the similarity measure. To implement the networks, we use Keras (Chollet, 2015).

As we show in the Section 4, these different configurations of siamese networks obtained poor performances over the Trial dataset and for that reason we did not submit their results to the ASSIN competition.

4 Evaluation Results

In this section, we discuss the results obtained with the methods described in Section 3. For such evaluation, we consider the Trial dataset as test set, and both PT-BR and PT-PT training sets. Note that we have removed from PT-BR the samples that also appear in Trial.

A comparison of the results for each method is presented in Table 1. In this case, the best results have been achieved with Kenter's features with either SVRs or Lasso for semantic similarity evaluation, and SVMs for entailment recognition. With SVR, Pearson correlation of 0.51, 0.49, and 0.50 have been reached for PT-BR, PT-PT, and Overall sets, respectively. In the entailment recognition task, F1 scores of 0.45, 0.50, and 0.51, have been achieved on the same sets, respectively. In addition, we observe that with Lasso, the results are very similar to those of SVR.

The second strategy, making use of Siamese networks, has not achieved good results. In the best result, LSTM obtained Pearson correlation of 0.41 using PT-BR as training data, which is 11 points below our best strategy. For this reason, we decided to submit the results only with Kenter's features, one run with SVR and another run with Lasso for semantic similarity, and one run with SVM in entailment recognition.

5 Competition Results

In this section we discuss the results of our best methods in the blind test data, and how it compared with the other competitors.

In total, six teams participated in the competition. In addition to our team, only two other teams submitted results for both tasks and both PT-BR and PT-PT sets. From the remaining three teams, two have focused only on the semantic similarity task, considering both sets, and the other one only on the PT-PT set, for both similarity and entailment recognition tasks.

The best result of each team[4], i.e. the best run, is listed in Table 2, and the ranking of each team, considering only the best run, is presented in Table 3. Considering only the best run of each team, we have managed to achieve very good results with the PT-BR set and Overall, being far from the first place only in the PT-PT set. With PT-BR, we ranked first in both accuracy and F1 metrics for entailment recognition, and second best in semantic similarity evaluation. Besides the good results, it was surprising that Kenter's features performed better in entailment recognition than semantic similarity evaluation, since the feature set has been originally proposed for the latter task. Overall, we ranked first in entailment recognition in F1, and second in accuracy. In semantic similarity, our team presented the second best Person correlation values, and the third best MSE value. In the PT-BR set, we have been able to be ranked second in F1 for the entailment recognition, and third in accuracy. But for semantic similarity, only the fourth place (tied with another team) has been reached.

One observation that is worth mentioning, is that in some tasks or sets the teams that achieved the best results were those that focused only on one task or set. For instance, the *Solo Queue* team submitted results only for semantic similarity, and they won the task for PT-BR and Overall, and ranked second for PT-PT. The *L2F/INESC-ID* team, on the other hand, submitted results only for PT-PT, for both tasks, and they won both. In our case, we submitted a single method, with almost no difference from one set to another or from one task to another, apart from the training sets. As lessons learned, in a future competition, we believe we shall invest more on fine tuning the algorithms to specific tasks and sets.

6 Conclusions and Future Work

In this paper we presented the methods and results followed by our team in the ASSIN competition, and evaluated the results obtained compa-

[4]Each team was allowed to submit up to three different runs

Tabela 1: Evaluation results (pearson correlation), considering Trial as test set.

Configuration	**Similarity**	**Entailment**
Baseline: Bag of Words Overall	0.47	
Kenter's features - SVR(M) PT-BR	0.51	79.60/0.45
Kenter's features - SVR(M) PT-PT	0.49	74.20/0.50
Kenter's features - SVR(M) Overall	0.50	77.00/0.51
Kenter's features - Lasso PT-BR	0.52	
Kenter's features - Lasso PT-PT	0.50	
Kenter's features - Lasso Overall	0.52	
CNN - PT-BR	0.35	
LSTM - PT-BR	0.41	

Tabela 2: Best results of each team in the competition.

	PT-BR				**PT-PT**				**Overall**			
	Sim		**RTE**		**Sim**		**RTE**		**Sim**		**RTE**	
Team	**P**	**MSE**	**Acc**	**F1**	**P**	**MSE**	**Acc**	**F1**	**P**	**MSE**	**Acc**	**F1**
Solo Queue	0.70	0.38	-	-	0.70	0.66	-	-	0.68	0.52	-	-
Reciclagem	0.59	1.31	79.05	0.39	0.54	1.10	73.10	0.43	0.54	1.23	75.58	0.40
ASAPP	0.65	0.44	81.65	0.47	0.68	0.70	78.90	0.58	0.65	0.58	80.23	0.54
LEC-UNIFOR	0.62	0.47	-	-	0.64	0.72	-	-	0.62	0.59	-	-
L2F/INESC-ID	-	-	-	-	0.73	0.61	83.85	0.70	-	-	-	-
Blue Man Group	0.65	0.44	81.65	0.52	0.64	0.72	77.60	0.61	0.63	0.59	79.62	0.58

red with the other teams. In our case, we decided to exploit word-vector-based approaches, following two distinct strategies: the first strategy is based on regular regression models that use a state-of-the-art feature set for semantic similarity encoding; and the second one is based on neural networks. Given the bad results of the second strategy in the evaluation datasets, we pursued in the competition only the method from the first strategy. With this approach, we have been ranked best in the entailment recognition task and in semantic similarity evaluation, achieving the best F1 score overall, and the best accuracy and F1 score in the PT-BR dataset. In the semantic similarity, our best result was the second place in the PT-BR set.

The experience of participating in the competition has been very valuable, and we expect to continue working on those problems to improve our methods and results. One future work is to understand better why siamese networks have not perform as well as the first strategy in these problems. Also, we would like to better investigate Kenter's features, in order to improve this feature set on these tasks.

Referências

2011. PASCAL recognizing textual entailment challenge (rte-7) at tac 2011. `http://www.nist.gov/tac/2011/RTE/`. Accessed: 2016-04-26.

Byun, Hyeran e Seong-Whan Lee. 2002. Applications of support vector machines for pattern recognition: A survey. Em *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, SVM '02, pp. 213–236, London, UK, UK. Springer-Verlag.

Chollet, François. 2015. Keras: Theano-based deep learning library. *Code: https://github.com/fchollet. Documentation: http://keras.io.*

Chopra, Sumit, Raia Hadsell, e Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 539–546. IEEE.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, e P. Kuksa. 2011. Natural language processing (almost) from

Tabela 3: Teams ranking considering the best run.

	PT-BR				**PT-PT**				**Overall**			
	Sim		**RTE**		**Sim**		**RTE**		**Sim**		**RTE**	
Team	**P**	**MSE**	**Acc**	**F1**	**P**	**MSE**	**Acc**	**F1**	**P**	**MSE**	**Acc**	**F1**
Solo Queue	1st	1st	-	-	2nd	2nd	-	-	1st	1st	-	-
Reciclagem	5th	5th	3rd	3rd	6th	6th	4th	4th	5th	5th	3rd	3rd
ASAPP	2nd	2nd	1st	2nd	3rd	3rd	2nd	3rd	2nd	2nd	1st	2nd
LEC-UNIFOR	4th	4th	-	-	4th	4th	-	-	4th	3rd	-	-
L2F/INESC-ID	-	-	-	-	1st	1st	1st	1st	-	-	-	-
Blue Man Group	2nd	2nd	1st	1st	4th	4th	3rd	2nd	2nd	3rd	2nd	1st

scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Cortes, Corinna e Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September, 1995.

Hochreiter, Sepp e Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kenter, Tom e Maarten de Rijke. 2015. Short text similarity with word embeddings. Em *CIKM 2015: 24th ACM Conference on Information and Knowledge Management*. ACM, October, 2015.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288.