LIPN-IIMAS at SemEval-2016 Task 1: Random Forest Regression Experiments on Align-and-Differentiate and Word Embeddings penalizing strategies

Oscar Lithgow
Centro de Ciencias Genómicas (CCG)
Universidad Nacional Autónoma de México (UNAM)
Ciudad Universitaria, DF, Mexico
owl@turing.iimas.unam.mx

Ivan V. Meza, Albert Orozco
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Universidad Nacional Autónoma de México (UNAM)
Ciudad Universitaria, DF, Mexico
{ivanvladimir,albert}@turing.iimas.unam.mx

Jorge Gacia Flores, Davide Buscaldi
Laboratoire d'Informatique de Paris Nord, CNRS (UMR 7030)
Université Paris 13, Sorbonne Paris Cité, Villetaneuse, France
{buscaldi,jgflores}@lipn.univ-paris13.fr

Abstract

This paper describes the *SOPA-N* system used by the LIPN-IIMAS team in Semeval 2016 Semantic Textual Similarity (Task 1). We based our work on the *SOPA 2015* system. The *SOPA-2015* system used 16 similarity features (including Wordnet, Information Retrieval and Syntactic Dependencies) within a Random Forest learning model. We expanded this system with an Align and Differentiate based strategy, word embeddings and penalization, which showed 6.8% of improvement on the development set. However, we found that on the evaluation data for the 2016 STS shared task, the 2015 system outperformed our newer systems.

1 Introduction

The *SOPA* system combines a regression model with multi-level similarity measures, from very simple ones (like edit distance) to more sophisticated ones (like IR-based or Wordnet similarity) (Buscaldi et al., 2013). Our goal this year was to add an align-and-differentiate penalizing strategy based on (Han et al., 2015) to our 2015 system (*SOPA*) (Buscaldi et al., 2015). The previous version consists on 16 similarity features which are regressed using a Random Forest learning algorithm. The rationale of the penalization was to account for cases when apparent distributional alignments are closer than their semantically equivalent (for instance, colors: while *black* and *white* are distributional close because are colors, but they are not semantically equivalent). We present two versions of the enhanced system SOPA 100 and 1000 which refers to the number of estimation trees used by the Random Forest algorithm.

We find that augmenting our 2015 model with an align and differentiate module boosts performance on the 2015 evaluation data. However, on the STS 2016 test data it only outperformed our previous approach on the plagiarism dataset. A closer error analysis showed that the gap between *SOPA* 100 and the gold standard (GS) scores are systematically positive, meaning that our new system overestimated the semantic similarity between phrases. Another interesting finding from the error analysis was that in those sentences where *SOPA* 100 outperform *SOPA*, it was also more accurate, given the fact that the *SOPA* 100 standard deviation from the gold standard annotation was smaller than the one of

Proceedings of SemEval-2016, pages 726–731,
San Diego, California, June 16-17, 2016.

SOPA.

Ablation tests for every feature were performed as well. Within our 2015 system, Sultan's alignment based similarity feature, Sultan et al. (2015)'s feature 18 , seems to be pulling down scores both for the headlines and question-question domains of 2016 dataset, despite the fact that 2015 train and test sets were used to train our 2016 system. Further analysis might be necessary in order to fully answer why *SOPA* 100 runs seemed to improve our scores for evaluation data from 2015, but they didn't for 2016 datasets.

2 SOPA

The SOPA system was built upon the idea on combining simple measures with a regression model to obtain a global, graded measure of textual similarity. Past experiences on this system have shown that the Random Forest (Breiman, 2001) outperforms other regression algorithms (like ν-Support Vector Regression or Multi-Layer Perceptron). Table 1 shows individual text similarity measures used as features for the global system, which have been described in more detail in (Buscaldi et al., 2013) and (Buscaldi et al., 2015).

	Measure
1	N-gram Based Similarity
2	WordNet Conceptual Similarity (Wu Palmer)
3	Syntactic Dependencies
4	Edit Distance
5	Cosine tf-idf
6	Named Entity Overlap
7	WordNet Conceptual Similarity (Jiang-Corath)
8	Information Retrieval Similarity (AQUAINT)
9	Geographical Context Similarity
10	Rank-Biased Overlap Similarity
11	DBPedia named entity Similarity
12	IR-based similarity (UkWaC index)
13	Skip-gram similarity
14	Sphynx WER
15	Sultan Similarity
16	Sentence size similarity
17*	Sultan alignment with word2vec* (left-right)
18*	Sultan alignment with WordNet (left-right)
19*	Sultan alignment with word2vec* (right-left)
20*	Sultan alignment with WordNet (right-left)
21*	Word2vec simple alignment (left-right)
22*	Word2vec simple alignment (right-left)
23*	Average of word2vec alignments

Table 1: Similarity Measures (* new features for 2016.)

2.1 Align-and-differentiate

This year we included a strategy that consists in two main steps:

1. Align words of both phrases
2. Differentiate or penalize those alignments that distributionally appear closer than they really are.

In the first step, the first word of phrase A is compared against each word of phrase B, and the pair which has the best similarity score is considered as an alignment pair and the words are removed from candidates. Then the process is repeated for the second word of phrase A and so on until all words are used.

The second step consists in inspecting each candidate alignment and penalizing those alignments that do not represent interchangeable concepts (synonyms). First, if an alignment score does not surpass a threshold, the alignment pair is discarded and those words are considered out-of-context. Second, the words of each pair is searched in Wordnet and: if those words are antonyms the alignment is penalized; if both words are hyponyms of a Disjoint Similar Concept (DSC) (Han et al., 2015), the alignment is penalized as well.

We also included an alignment-shift-penalization. It is computed using Spearman correlation over the position in the phrase of the aligned words. The idea behind is to reduce the similarity score on those phrase alignments in which aligned pairs consist in words with a very different position in phrase A with respect phrase B.

2.2 Alignments

Two different alignment strategies were tried. A pure distributional based alignment in which words are aligned using *word2vec* (Mikolov et al., 2013) to measure the cosine similarity between the two words and aligned pairs are determined using the best local alignment. In this approach, a left-right alignment and a right-left alignment are combined in order to,

in some way, alleviate the appearance of local maxima.

The second is a hybrid strategy that combines a syntactic based alignment produced by Sultan with a distributional alignment. This is, it uses the pairs given by Sultan aligner (Sultan et al., 2014) and those not aligned words of both phrases are submitted to a second alignment process (distributional) that in some sense complements the syntactic alignment.

3 Experimental setting

We used the following configurations for *SOPA* 100 and *SOPA* 1,000 runs, they were applied to the features 17-23:

- Words identification and tokenization was done using a simple space-based regular expression.
- Similarity in WordNet was obtained using the *path_similarity* metric available in NLTK package (Loper and Bird, 2002).
- For the penalization by shift-external-alignment, the Spearman correlation between the positions of aligned words is computed and, if the correlation is above a threshold of 0.25, it is multiplied by the alignment score.
- When checking the aligned pairs of words in WordNet: if one word of the pair is listed as an antonym of the other, a penalty of 1.0 is applied.; if words are not antonyms, we search the lowest common hypernym for both synsets, then we walk the path from this hypernym to the root and if in this walk we visit one of the nodes considered DSC then a penalty of 0.5 is applied.
- We used cosine as the measure of distance among vectors. The vectors are extracted form the standard Google pretrained *word2vec* models with 300 dimensions.[1]
- We used the list of English stop-words available in the NLTK package to filter stop-words from sentences.
- We discarded alignments that are below an align-threshold of 0.3.
- Each unaligned word was considered out-of-context (OOC) and for each of those, a penalization of 1.0 was applied.

The code of our implementation and experimentation is openly available.[2]

4 Results

These year runs were:

SOPA The *SOPA* 2015 system (16 scores) trained with a random forest regressor set to 100 estimators (Buscaldi et al., 2015).

SOPA 100 This was the *SOPA* similarities plus the align-and-differentiate scores trained with a random forest regressor set to 100 estimators.

SOPA 1000 This was the *SOPA* similarities plus the align-and-differentiate features trained with a random forest regressor set to 1,000 estimators.

Table 2 shows the performance of each of these systems on the 2016 STS evaluation data. As it can be appreciated the *SOPA* outperforms the other runs in most domains and in the overall evaluation. The extracted scores using the align-and-differentiate strategy were not helpful for these datasets. This was contrary to our development experience in which the these scores contributed to improve the performance of the 2015 *SOPA* system (see Table 3 with 2015 test dataset).

The change of behavior from the developing to the testing stage was unexpected. With this in mind we proceed to do further experimentation. First we re-run the *SOPA* 100 and *SOPA* 1,000 setting but only using the align-and-differentiate scores. Table 4 present these results, as expected this time the performance is quite poor. Effectively, the performances of the score were too poor, so that they bring down the full performance. These experiments confirms what we learn from Table 2. In order to gain a better insight on the scores we run an ablation experiment. In this case we suppress each the score and measure the loss in performance. Table 5 presents

[1] https://code.google.com/archive/p/word2vec/

[2] https://github.com/rcln/SemEval

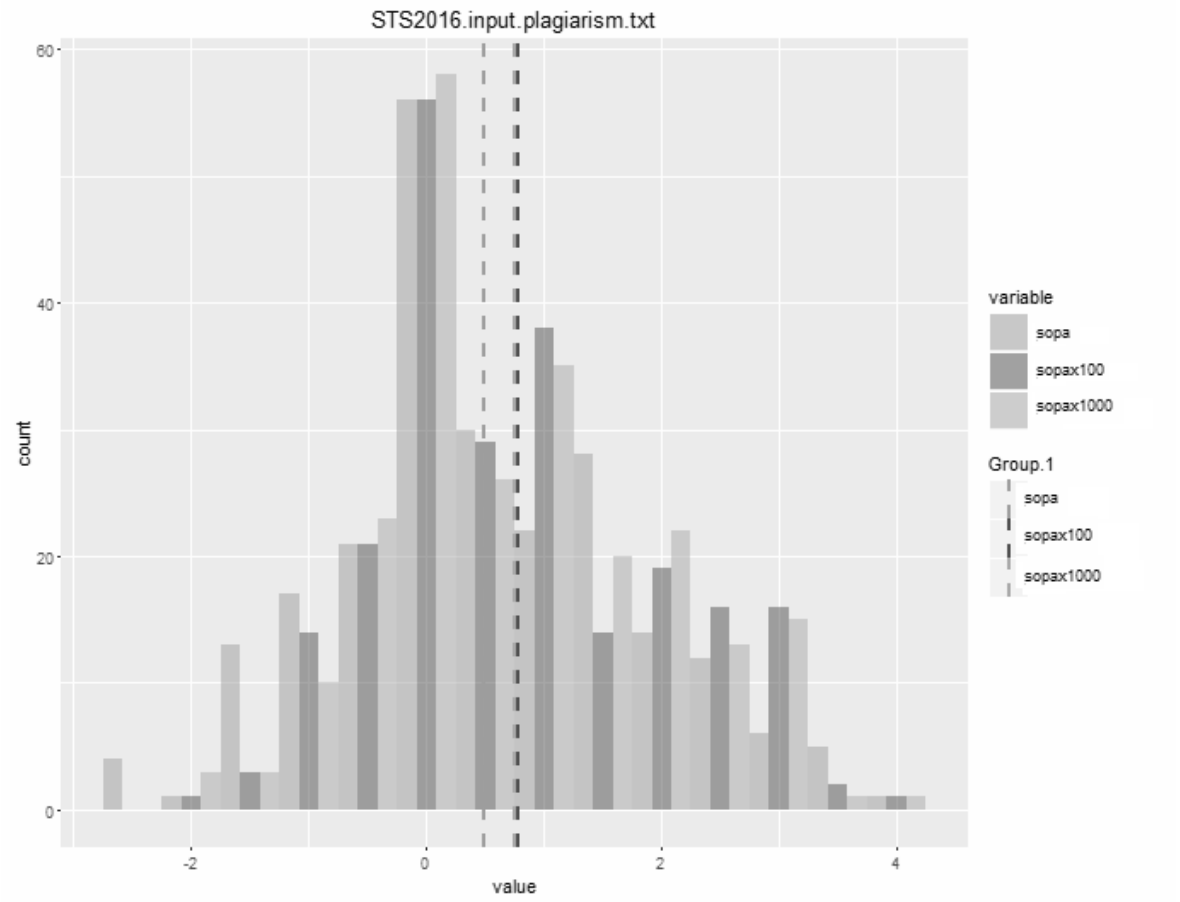

Figure 1: Distribution of the difference with plagiarism

these losses. The table marks with a star those score which have a negative loss, meaning that without them the performance will increase. However, we do not observe a general pattern. Similarity scores for some domains made them particularly worse, but not for others. The *headlines*, *plagiarism* and *question-question* has a larger amount of features that were not very useful. We identify 8 scores with a negative overall loss, however further experimentation without these features did not give a better performance for the *SOPA* 100 and *SOPA* 1,000 settings.

Figure 2 shows the differences of the runs with the gold standard. The first thing to notice from this graph is that all systems tend to overestimate the score, this is they give a larger score than the gold standard. We also notice that the standard deviation is close to one for most of the runs. However, these graphs do not tell the whole story, for instance in the case of the *plagiarism* domain the distribution of the difference is far from normal, Figure 1 shows the histogram of score differences, illustrating a slight positive skew in the predictions.

Dataset	SOPA	100	1,000
answer-answer	**0.44901**	0.43216	0.44893
headlines	**0.62411**	0.58499	0.59721
plagiarism	0.69109	0.74727	**0.75936**
postediting	**0.79864**	0.75560	0.76157
question-question	**0.59779**	0.55310	0.56285
Overall	**0.63087**	0.61321	0.62466

Table 2: Final test results

5 Conclusions and perspectives

In this paper we introduced the *SOPA-N* system used to calculate semantic similarity between sentence pairs for the Semeval STS 2016 Challenge. *SOPA-N* is based in a 23 similarity feature set and

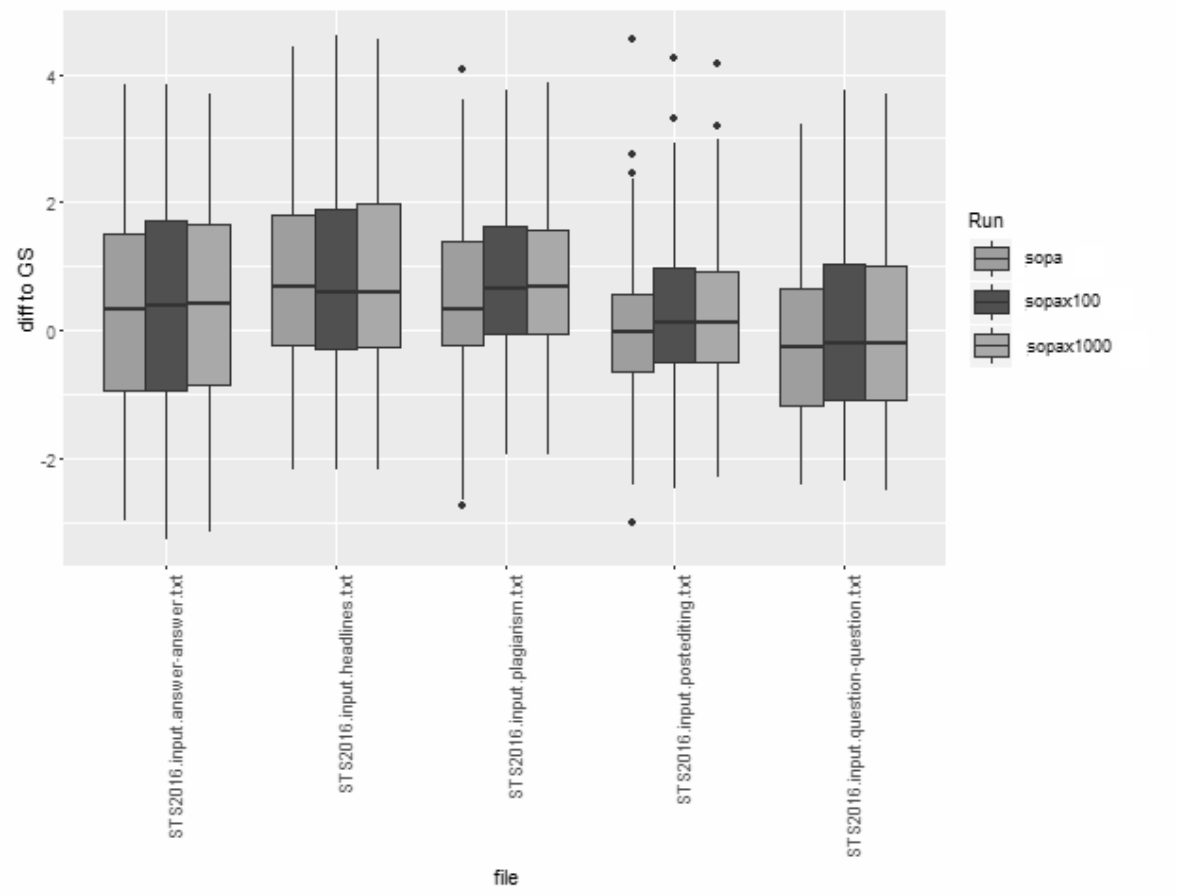

Figure 2: Distribution of the difference with GS for the plagiarism corpus

Dataset	SOPA	100	1000
answer-forums	0.63331	0.64814	**0.65398**
answers-students	0.63893	**0.70356**	0.70137
images	0.81768	0.84109	**0.84237**
headlines	0.81666	**0.8247**	0.82376
believe	0.59779	0.73149	**0.73179**
Overall	0.68232	0.74976	**0.75065**

Table 3: Development results (STS 2015)

Dataset	100	1000
answer-answer	0.40169	0.39145
headlines	0.50688	0.50928
plagiarism	0.70305	0.70814
postediting	0.6669	0.66366
question-question	0.37848	0.39587

Table 4: Re-run of **100** system with only align-and-differentiate scores

a Random Forest learning algorithm. New features for this year challenge included word embeddings based on Sultan et al. (2015)'s alignment and an Align-And-Differentiate strategy inspired by Han et al. (2015). These improvements allowed us to outperform our previous approach only for the *plagiarism* test corpus. Even if *SOPA-N* was more accurate for those sentences where its score gap between the gold standard was better than *SOPA*, further error analysis is needed.

We hypothesize that the addition of align based features was not appropriate for this year data. Particularly we plan to analyze alignment errors and expand the context to better guide the alignment. Also, future research will focus in a deeper characterization of 2016 test in terms of spelling correctness, typography profile POS-tag categories, semantic and discursive level in order to find correlation between these characteristics and the GS gap of our system.

	a.-a.	head.	pla.	post.	q.-q.
1	−0.01	−0.02	0.02	0.00	0.02
2	0.01	**-0.04**	0.02	−0.01	0.02
3*	0.00	**-0.03**	−0.02	0.01	−0.01
4	0.00	0.00	0.01	0.00	−0.01
5	0.02	0.00	0.01	0.00	−0.02
6	0.02	0.00	0.00	0.00	0.00
7	0.01	0.02	0.00	0.00	0.01
8	0.00	0.01	−0.01	0.00	0.01
9	0.00	0.00	0.02	0.00	0.02
10*	0.01	−0.02	−0.02	0.00	−0.01
11*	0.01	−0.02	−0.01	−0.01	0.01
12*	0.00	−0.01	0.00	0.00	−0.02
13*	0.01	0.01	−0.01	−0.01	−0.01
14	0.00	0.01	0.00	0.01	0.01
15*	0.00	**-0.04**	−0.01	0.00	**-0.05**
16	0.01	−0.01	0.01	0.02	−0.00
17*	0.01	0.00	0.01	0.01	**0.03**
18	0.01	0.01	−0.01	0.01	**-0.04**
19	0.02	0.01	0.01	0.01	0.01
20*	0.01	**-0.03**	0.00	0.00	−0.01
21	0.01	0.00	−0.02	0.00	0.02
22	0.02	0.01	0.00	0.00	0.02
23	0.00	−0.02	**0.03**	0.00	0.02

Table 5: Loss for each score for **100** run

6 Acknowledgments

Work by Oscar Lithgow was supported by the National Institute of Health (Award number: GM110597) through the Centro de Ciencias Computacionales, UNAM.

References

[Breiman2001] Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

[Buscaldi et al.2013] Davide Buscaldi, Joseph Le Roux, Jorge J. Garcia Flores, and Adrian Popescu. 2013. LIPN-CORE: Semantic Text Similarity using n-grams, WordNet, Syntactic Analysis, ESA and Information Retrieval based Features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 162–168, Atlanta, Georgia, USA, June.

[Buscaldi et al.2015] Davide Buscaldi, Jorge García Flores, Ivan V. Meza, and Isaac Rodríguez. 2015. SOPA: Random Forests Regression for the Semantic Textual Similarity task. In *SemEval 2015*, pages 132–137.

[Han et al.2015] Lushan Han, Justin Martineau, Doreen Cheng, and Christopher Thomas. 2015. Samsung: Align-and-differentiate approach to semantic textual similarity. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 172–177, Denver, Colorado, June. Association for Computational Linguistics.

[Loper and Bird2002] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

[Sultan et al.2014] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to Basics for Monolingual Alignment: Exploiting Word Similarity and Contextual Evidence.

[Sultan et al.2015] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Dls@cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153, Denver, Colorado, June. Association for Computational Linguistics.