UdL at SemEval-2017 Task 1: Semantic Textual Similarity Estimation of English Sentence Pairs Using Regression Model over Pairwise Features

Hussein T. Al-Natsheh [123], **Lucie Martinet** [124], **Fabrice Muhlenbach** [15], **Djamel A. Zighed** [12]
[1]Université de Lyon,
[2]Lyon 2, ERIC EA 3083, 5 Avenue Pierre Mendès France, F69676 Bron Cedex - France
[3]CNRS, ISH FRE 3768, 14 Avenue Berthelot, F-69007 Lyon, France
[4]CESI EXIA/LINEACT, 19 Avenue Guy de Collongue, F-69130 Écully, France
[5]UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516, F-42023 Saint Etienne, France
hussein.al-natsheh@cnrs.fr, lucie.martinet@eric.univ-lyon2.fr
fabrice.muhlenbach@univ-st-etienne.fr, djamel@zighed.com

Abstract

This paper describes the model UdL we proposed to solve the semantic textual similarity task of SemEval 2017 workshop. The track we participated in was estimating the semantics relatedness of a given set of sentence pairs in English. The best run out of three submitted runs of our model achieved a Pearson correlation score of 0.8004 compared to a hidden human annotation of 250 pairs. We used random forest ensemble learning to map an expandable set of extracted pairwise features into a semantic similarity estimated value bounded between 0 and 5. Most of these features were calculated using word embedding vectors similarity to align Part of Speech (PoS) and Name Entities (NE) tagged tokens of each sentence pair. Among other pairwise features, we experimented a classical tf–idf weighted Bag of Words (BoW) vector model but with character-based range of n-grams instead of words. This sentence vector BoW-based feature gave a relatively high importance value percentage in the feature importances analysis of the ensemble learning.

1 Introduction

Semantic Textual Similarity (STS) is a shared task that have been running every year by SemEval workshop since 2012. Each year, the participating teams are encouraged to utilize the previous years data sets as a training set for their models. The teams are then ranked by their test score on a hidden human annotated pairs of sentences. After the end of the competition, the organizers publish the gold standards and ask the teams of the coming year task to use it as a training set and so on. The description of STS2017 task is reported in (Cer et al., 2017). In STS2017 , the primary task consisted in 6 tracks covering both monolingual and cross-lingual sentence pairs for the languages Spanish, English, Arabic, and Turkish. Our team, UdL, only participated in the English monolingual track (Track 5).

The data consist in thousands of pairs of sentences from various resources like (Twitter news, image captions, news headline, questions, answers, paraphrasing, post-editing...). For each pair, a human annotated score (from 0 to 5) is assigned and indicates the semantic similarity values of the two sentences. The challenge is then to estimate the semantic similarity of 250 sentence pairs with hidden similarity values. The quality of the proposed models would then be evaluated by the Pearson correlation between the estimated and the human annotated hidden values.

In section 2, we link to some related work to this problem. The data preparation method followed by a full description of the model pipeline and its implementation are then presented in sections 3, 4, and 5. Results of the model selection experiments and the final task results are shown in section 6.

2 Related Work

The general description of the methodologies proposed by the task previous year winners are discussed in Agirre et al. (2016). However, there were many other related work to solve the issue of encoding semantics of short text, i.e., sentences or paragraphs. Many of them tend to reuse word embeddings (Pennington et al., 2014) as an input for sentence embedding, while others (Shen et al., 2014; Le and Mikolov, 2014) propose to directly learn the sentence semantics features. Most of these embedding techniques are based on large

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 115–119,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

text corpus where each word or short text dense vector representations (i.e., word embedding) are learned from the co-occurrence frequencies with other words in the context. Other methodologies are based on matrix decomposition of the Bag of Word (BoW) matrix using Latent Semantic Analysis (LSA) techniques like Singular Value Decomposition (SVD) or Non-Negative Matrix Factorization (NMF). According to a comparable transfer learning strategy (Bottou, 2014), if we are able to build a model consisting in (1) a pairwise transformer (i.e., feature extractor), and (2) a comparator that can well-predict if the two elements of the input are of the same class or not, then the learned transformer could be reused to easily train a classifier to label a single element. A good example to understand such system is face recognition, e.g., it is considered impossible to have all human faces images to train the best features set of a face, however, a learned model that can tell if two given face-images are of the same person or not, could guide us to define a set of good representative features to recognize a person given one face image. We can generate $\frac{2^n}{2}$ comparative pairs from n examples. Similarly, we cannot have all possible sentences to identify the sentence semantics, but we can generate a lot of comparative sentence pairs to learn the best semantics features set, i.e., sentence dense vector representation. Thus we consider our pairwise feature-based model as an initial step to build a sentence dense vector semantics representation that can perform very well in many applications like semantics highlighter, question answering system and semantics-based information retrieval system.

3 Data Set Preparation

The data set provided for the STS task consists in a set of tab-separated values data files from different text types accommodated year-after-year since 2012. Each year, the task organizers provide additional data files from different text sources. The text sources vary between classical narrative sentences, news headlines, image captions or forum questions or even chat Twitter news. The source types used in the task are listed in Table 1.

Each files pair consists of a first file containing, at each line, the two sentences to be compared and some information about the sources of these sentences if any. The second file contains, at each line, the similarity score of the corresponding pair of sentences that is presented in the first file. In addition, for the data extracted from the previous years, we have one directory for the training set and another one for tests. We noticed that the separator format for the data file is not optimized since using a tabulator can make things confused because it is also a character used in some cases inside the text. This could be solved only by hand, after a first automatic preprocessing. After that, we can read the file by line, looking for the good characters and line format. We are also grateful that our predecessors, e.g., Tan et al. (2015), who shared some of their aggregated data that we could also add to our training set. In the end, we used the set of data of all the previous years since 2012. An additional step we considered was the spell-checker correction using *Enchant* software. We assume that such preprocessing step could enhance the results. However, this step was not used in our submitted system. Finally, we also consider a version of the data set where we filtered out the hash-tag symbol from the Twitter news sentence pairs.

4 Model Description

Our approach is based on the comparable transfer learning systems discussed in section 2. Accordingly, our model pipeline mainly consists in 2 phases: (1) pairwise feature extraction, i.e., feature transformer, and (2) regression estimator. While many related work either use words embedding as an input for learning the sentence semantics representation or learning such semantics features directly, our model is able to reuse both types as input for the pairwise feature transformer. For example, as listed in Table 2, we used features that is based on word vectors similarity of aligned words while we also have a feature that consider the whole sentence vector, i.e., sparse BoW. The model can also use, but not yet used in this paper, unsupervised learned sentence representation out of methods like BoW *matrix decomposition*, *paragraph vector*, or *sent2vec* methods as input to our pairwise features transformer.

4.1 Pairwise Feature Extraction

We used different feature types as in Table 2. The first two types are based on aligning PoS and NE tagged words and then compute the average word vectors cosine similarity (CS) of the paired tags. The process of extracting these type of pairwise

Source Types (as named in the source file)	Manually Assigned Domain Class
FNWN, OnWN, surprise.OnWN	Definition
MSRpar, belief, plagiarism, postediting	Paraphrasing
MSRvid, images	Image-captions
SMT, SMTeuroparl, deft-news, headlines, surprise.SMTnews, tweet-news	News
answer-answer, answers-forums, answers-students, deft-forum, question-question	Question-answer

Table 1: Sentence pairs data source types and its manually annotated domain class.

Algorithm 1: The pairwise features extraction process of aligned PoS and NE tagged tokens.

```
Input: Sentence pair
Extract a PoS type or a NE type word tokens
  from both sentences
Pair each tagged word-token in one sentence
  to all same tagged tokens in the other
  sentence
Get the word vector representations of both
  tokens of each paired tokens
Compute the vector representations of both
  tokens of each paired tokens
Align words if the cosine similarity (CS) is
  above a threshold value
Solve alignment conflicts, if any, based on the
  higher CS value
Compute the average CS of the aligned tokens
  and use it as the pairwised feature value
```

features are resumed in the algorithm 1.

The third feature is extracted by transforming each sentence to its BoW vector representation. This sparse vector representation is weighted by tf–idf. The vocabulary of the BoW is the character grams range between 2 and 3. This BoW vocabulary source is only the data set of the task itself and not a general large text corpus like the ones usually used for word embedding. We are planning to try out a similar feature, but unsupervised, where we consider a corpus like Wikipedia dump as a source for the BoW. Another feature we plan to consider as a future work is the dense decomposed BoW using SVD or NMF. Finally, we can also consider unsupervised sentence vectors using *paragraph vectors* or *sent2vec* methods.

Features number 4 is extracted by computing the absolute difference of the summation of all numbers in each sentence. To achieve that, we transferred any spelled number, e.g., “sixty-five”, to its numerical value, e.g., 65. The fifth pairwise feature we used was simply based on the sentence length. The last feature is extracted by mapping each sentence pair source to a manually annotated domain class as in Table 1. However, in order to use this feature, we would need to specify the domain class of the sentence pairs of the test data set. Manually checking the test data and also based on some replies found from the task organizers about the source of the test data, we classified them all as “Image-captions”.

4.2 Regression

We have mainly evaluated two regression estimators for this task. The first estimator was random forests (RF) and the other was Lasso (least absolute shrinkage and selection operator). Based on a 10-fold cross-validation (CV), we set the number of estimators of 1024 for RF and a maximum depth of 8. For Lasso CV, we finally set the number of iterations to 512.

5 Implementation

Our Python model implementation is available for reproducing the results on GitHub[1]. For PoS and RE tagging, we utilized both polyglot (Al-Rfou et al., 2013) and spaCy. We used a pre-trained GloVe (Pennington et al., 2014) word vectors of a size 300. The pipeline of the transformer and the regression estimator was built on scikit-learn API. Finally, we used pair-wise feature combiners similar to the ones used in Louppe et al. (2016).

6 Results

6.1 Regression Estimator Selection

First, we run few experiments to decide on using RF or Lasso CV. The experimental results of these runs are listed in Table 3. The feature-importances analysis are shown in the right column of Table 2.

[1] https://github.com/natsheh/sensim

	Feature	Pair Combiner	Importance
1	Aligned PoS tags (17 tags)	Average of w2v CS of all PoS tag pairs	0.113
2	Aligned NE tags (10 tags)	Average of w2v CS of all NE tag pairs	0.003
3	TFIDF char ngrams BoW	Cosine similarity of the sentence BoW vector pair	0.847
4	Numbers	Absolute difference of the number summation	0.006
5	Sentence length	Absolute difference of the number of characters	0.032
6	Domain class of the pair	N/A	N/A

Table 2: Pairwise features set.

Regressor	PoS	word_vectors	images	answers_students	headlines_2016	Mean
Lasso CV	polyglot	GloVe	0.82	0.74	0.80	0.79
Lasso CV	spaCy	spaCy	0.82	0.74	0.79	0.79
RF	spaCy	spaCy	0.85	0.78	0.80	0.81
RF	polyglot	spaCy	0.85	0.77	0.80	0.81

Table 3: Regression estimator selection based on experimental evaluation score over a few data sets.

6.2 System Configuration Selection

We experimented different settings varying the feature transformation design parameters and trying out three different training set versions for RF. We show the 3 selected settings for submission and the test score of a few evaluation data-sets from previous years in Table 4.

6.3 Final Results

We finally submitted three runs of our model UdL for the task official evaluation. The settings of these three runs are shown in Table 4. The summary of the evaluation score with the baseline (0.7278), the best score run model (0.8547), the least (0.0069), the median (0.7775) and the mean (0.7082) are shown in Figure 1. Run1 was our best run with Pearson correlation score of (0.8004), At this run, we used RF for regression estimator on our all extracted pairwise features except the domain class feature. Run2 (0.7805) was same as Run1 except that we used the domain class feature. Finally, Run3, submission correction phase (0.7901), used a different data set were we filtered-out hash-tag symbol from Twitter-news sentence pairs.

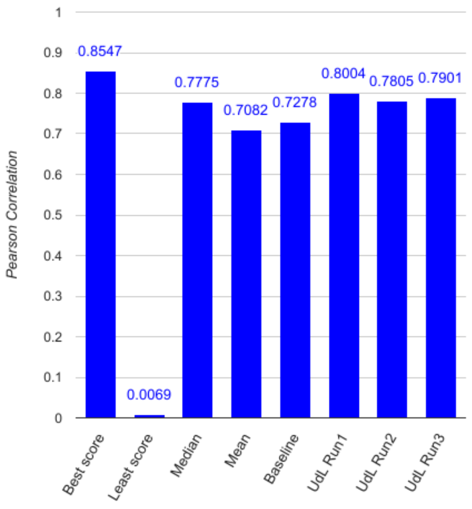

Figure 1: Track 5 results summary in comparison to UdL three runs;*: submission correction.

7 Conclusion and Future Work

We proposed UdL, a model for estimating sentence pair semantic similarity. The model mainly utilizes two types of pairwise features which are (1) the aligned part-of-speech and named-entities tags and (2) the tf–idf weighted BoW vector model of character-based n-gram range instead of words. The evaluation results shows that Random Forest regression estimator on our extracted pairwise features provided 80% of Pearson correlation with hidden human annotation values. The model was implemented in a scalable pipeline architecture and is now made available to the public where the user can add and experiment any additional features or even any other regression models. Since the sentence vector BoW-based pairwise feature showed high percentage in the feature importances analysis of the Random Forest estimator, we are going to try other, but dense, sentence vector representation, e.g., in Shen et al. (2014); Le and Mikolov (2014). We are also planning to use and evaluate the model in some related applications in-

Submission	data set	DF	PoS	vectors	images	AS	H16	AA	QQ	plagiarism	mean
-	small	no	polyglot	spaCy	0.85	0.77	0.80	0.47	0.54	0.82	0.71
-	small	yes	polyglot	spaCy	0.82	0.75	0.79	0.53	0.56	0.84	0.72
Run2 settings	big	yes	spaCy	spaCy	0.82	0.74	0.79	0.54	0.61	0.84	0.72
-	big	yes	polyglot	spaCy	0.82	0.75	0.79	0.52	0.55	0.84	0.71
-	big	no	spaCy	spaCy	0.82	0.78	0.80	0.46	0.60	0.82	0.71
-	big	no	polyglot	spaCy	0.85	0.77	0.80	0.51	0.56	0.82	0.72
Run1 settings	big	no	polyglot	spaCy	0.85	0.77	0.80	0.46	0.54	0.82	0.71
Run3 settings	BH	no	polyglot	spaCy	0.85	0.77	0.80	0.51	0.58	0.82	0.72
-	BH	no	polyglot	GloVe	0.85	0.77	0.80	0.46	0.57	0.81	0.71

Table 4: Evaluation 2-decimal-rounded score on some testsets. DF: domain feature, AA:answer-answer, AS:answers_students, H16:headlines_2016, QQ:question-question, BH:bigger data set size where hashtags are filtered

cluding a semantic sentences highlighter, a topic-diversified document recommender system as well as a question-answering system.

Acknowledgments

We would like to thank ARC6 Auvergne-Rhône-Alpes that funds the current PhD studies of the first author and the program "Investissements d'Avenir" ISTEX for funding the post-doctoral position of the second author.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Steven Bethard, Daniel Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 497–511.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In Julia Hockenmaier and Sebastian Riedel, editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*. ACL, pages 183–192.

Léon Bottou. 2014. From machine learning to machine reasoning - an essay. *Machine Learning* 94(2):133–149.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 1–14. http://www.aclweb.org/anthology/S17-2001.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. JMLR.org, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196.

Gilles Louppe, Hussein T. Al-Natsheh, Mateusz Susik, and Eamonn James Maguire. 2016. Ethnicity sensitive author disambiguation using semi-supervised learning. In Axel-Cyrille Ngonga Ngomo and Petr Kremen, editors, *Knowledge Engineering and Semantic Web - 7th International Conference, KESW 2016, Prague, Czech Republic, September 21-23, 2016, Proceedings*. Springer, volume 649 of *Communications in Computer and Information Science*, pages 272–287.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, pages 1532–1543.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. ACM, pages 101–110.

Liling Tan, Carolina Scarton, Lucia Specia, and Josef van Genabith. 2015. USAAR-SHEFFIELD: semantic textual similarity with deep regression and machine translation evaluation metrics. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*. The Association for Computer Linguistics, pages 85–89.