



Predicting the Semantic Textual Similarity with Siamese CNN and LSTM

Elvys Linhares Pontes, Stéphane Huet, Andréa Carneiro Linhares,
Juan-Manuel Torres-Moreno

► To cite this version:

Elvys Linhares Pontes, Stéphane Huet, Andréa Carneiro Linhares, Juan-Manuel Torres-Moreno. Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. Traitement Automatique des Langues Naturelles (TALN), May 2018, Rennes, France. pp.311-319. hal-01779457

HAL Id: hal-01779457

<https://hal.archives-ouvertes.fr/hal-01779457>

Submitted on 26 Apr 2018

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Predicting the Semantic Textual Similarity with Siamese CNN and LSTM

Elvys Linhares Pontes¹ Stéphane Huet¹ Andréa Carneiro Linhares²
Juan-Manuel Torres-Moreno^{1,3}

(1) LIA, Université d'Avignon et des Pays de Vaucluse, Avignon, 84000 France

(2) Universidade Federal do Ceará, Sobral, Ceará Brazil

(3) École Polytechnique de Montréal, Montréal, Canada

{elvys.linhares-pontes, juan-manuel.torres,
stephane.huet}@univ-avignon.fr

andrea.linhares@ufc.br

RÉSUMÉ

La Similarité Textuelle Sémantique (STS) est la base de nombreuses applications dans le Traitement Automatique du Langage Naturel (TALN). Notre système combine des réseaux neuronaux convolutifs et récurrents pour mesurer la similarité sémantique des phrases. Il utilise un réseau convolutif pour tenir compte du contexte local des mots et un LSTM pour prendre en considération le contexte global d'une phrase. Cette combinaison des réseaux préserve mieux les informations significatives des phrases et améliore le calcul de la similarité entre les phrases. Notre modèle a obtenu de bons résultats et est compétitif avec les meilleurs systèmes de l'état de l'art.

ABSTRACT

Semantic Textual Similarity (STS) is the basis of many applications in Natural Language Processing (NLP). Our system combines convolution and recurrent neural networks to measure the semantic similarity of sentences. It uses a convolution network to take account of the local context of words and an LSTM to consider the global context of sentences. This combination of networks helps to preserve the relevant information of sentences and improves the calculation of the similarity between sentences. Our model has achieved good results and is competitive with the best state-of-the-art systems.

MOTS-CLÉS : Similarité, Réseaux de neurones siamois, LSTM, CNN.

KEYWORDS: Similarity, Siamese Neural Networks, LSTM, CNN.

1 Introduction

Semantic Text Similarity (STS) is an important task in Natural Language Processing (NLP) applications such as information retrieval, classification, extraction, question answering, and plagiarism detection. The STS task measures the degree of similarity between two texts and can be expressed as follows: given two sentences, a system returns a continuous score on a scale from 1 to 5, with 1 indicating that the semantics of the sentences are completely independent and 5 meaning that there is a semantic equivalence.

STS is a difficult issue since languages have numerous ambiguities and synonymous expressions, while sentences may have variable lengths and complex structures. Therefore basic models, e.g.

bag-of-words or TF-IDF models, are constrained by their specificities that put aside the role played by the word order and ignore syntactic as well as semantic relationships. Recent successes in sentence similarity have been obtained using Neural Networks (RNNs: Recurrent Neural Networks (Mueller & Thyagarajan, 2016; Kiros *et al.*, 2015; Tai *et al.*, 2015) and CNNs: Convolutional Neural Networks (He *et al.*, 2015)). Neural Networks (NNs) use a deep analysis of sentences and words to take better into account both the semantics and the structure of sentences in order to predict the sentence similarity.

In this paper, we describe our technique based on NNs to measure similarity. First, we use a Siamese CNN to analyze the local context of words in a sentence and to generate a representation of the relevance of a word and its neighborhood. Then, we use a Siamese LSTM to analyze the entire sentence based on its words and its local contexts. At last, we predict the semantic similarity of pairs of sentences using the Manhattan distance.

We applied our framework on the SemEval information for STS assignment and we acquired competitive outcomes demonstrating that our model can give helpful information to enhance the sentence analysis.

This paper is organized as follows: we make an overview of relevant work for STS in Section 2. Next, we detail our approach in Section 3. The experimental setup and results are presented in Sections 4 and 5, respectively. Finally, we give our conclusion and some last remarks in Section 6.

2 Related Work

To deal with the STS task, previous studies have resorted to various features (e.g. word overlap, synonym/antonym), linguistic resources (e.g. WordNet and pre-trained word embeddings) and a wide assortment of learning algorithms (e.g. Support Vector Regression (SVR), regression functions and NNs). Among these works, several techniques extract multiple features of sentences and apply regression functions to estimate these similarity scores (Lai & Hockenmaier, 2014; Zhao *et al.*, 2014; Bjerva *et al.*, 2014; Severyn *et al.*, 2013). Lai & Hockenmaier (2014) analyzed distinctive word relations (e.g. synonyms, antonyms, and hyperonyms) with features based on counts of co-occurrences with other words and similarities between captions of images. Zhao *et al.* (2014) predicted the sentence similarity from syntactic relationship, distinctive content similitudes, length and string features. Bjerva *et al.* (2014) also utilized a regression algorithm to foresee the STS from different features (WordNet, word overlap, and so forth). Finally, Severyn *et al.* (2013) combined relational syntactic structures with SVR.

The development of NNs has improved the results of many NLP applications and especially the STS task (He *et al.*, 2015; Mueller & Thyagarajan, 2016; Tsubaki *et al.*, 2016; Rychalska *et al.*, 2016). Architectures such as RNNs and CNNs further improve the semantic analysis and the prediction of sentence relatedness.

RNNs differ from other NN models in their ability to process sequential information. They update a memory cell to make sense of data read in a sentence over time. Rychalska *et al.* (2016) used a Recursive AutoEncoder (RAE) and a WordNet graph framework to produce sentence embeddings. They consolidated these embeddings with a Support Vector Machine (SVM) classifier to compute a semantic relatedness score. Long Short Term Memory (LSTM) enhances RNNs to handle long-term dependencies (Mueller & Thyagarajan, 2016; Greff *et al.*, 2015; Tai *et al.*, 2015). The LSTM engineering is made out of a memory cell and non-direct gating units that update its state over time

and manage the data stream into/out the cell. Mueller & Thyagarajan (2016) used a Siamese LSTM to encode sentences using pre-trained word embedding vectors. Siamese LSTMs used the same weights to encode sentences and to produce comparable sentence representations for similar sentences. Then, they predicted the closeness of pair of sentences using the Manhattan distance between the sentence representations. Tai *et al.* (2015) introduced the Tree-LSTM that is a generalization of LSTM for tree-structured network topologies. They utilized this Tree-LSTM to encode a couple of sentences and to predict their closeness with a NN that analyzes the distance and the angle between the sentence embeddings.

CNNs have accomplished excellent outcomes in classification (Kim, 2014) and other NLP tasks (Collobert *et al.*, 2011). He *et al.* (2015) generated sentence embedding using a Siamese CNN architecture with various convolution and pooling operations to extract distinctive granularities of information. Their convolution uses filters that analyze entire word embeddings and each dimension of word embeddings with multiple window sizes. For output of the convolution operation, they applied several pooling types (max, mean, and min). Finally, they predicted the sentence similarity from numerous measurements (horizontal and vertical comparison) to compare local regions of sentence representation.

In this work, we join the ideas examined in (Mueller & Thyagarajan, 2016) and (Kim, 2014) to produce more accurate semantic sentence embeddings. The next section presents our model and its characteristics w.r.t. previous work.

3 Our model

A sentence is composed of words which can form phrases and clauses. Examining a sentence and its components helps us to comprehend its meaning. NNs are structures that can inspect relationships between words from multiple points of view. On the one hand, LSTMs can recognize and process the semantics of a sentence by investigating the words through time. They update their state to get the gist of the sentence (global context) in the order of words. In this procedure, LSTMs filter unimportant data by retaining just the main information. On the other hand, CNNs use layers with convolution filters that are connected to local features (Kim, 2014). They enable the analysis of a sentence from multiple perspectives (filters). This type of NNs does not have the same concern with the sentence length as LSTMs since CNNs examine all the words of the sentence together. Nonetheless, CNNs do not consider the order of words in their analysis, so these structures cannot investigate sequence relationships in the sentence.

Differently from (Mueller & Thyagarajan, 2016) that only analyze the general context of words and from (He *et al.*, 2015) that do not consider the order of words in the sentences, we analyze the words in two perspectives: general and local contexts. Words are considered through time from the general information of a word (word embedding) and its specific semantic and syntactic features (local context) based on its previous and its following words. We apply a CNN to investigate the local context for each word in a sentence. The CNN analyzes together all the words of the local context and generates their representation as a unique structure. Then, we utilize an LSTM to examine the words of the sentence one by one (Figure 1). Our NN has a Siamese structure (Mueller & Thyagarajan, 2016; He *et al.*, 2015), i.e. our CNN^A and our $LSTM^A$ are equal to our CNN^B and our $LSTM^B$, respectively. The following subsection describes our CNN, our LSTM, and our similarity metrics to predict the sentence similarity.

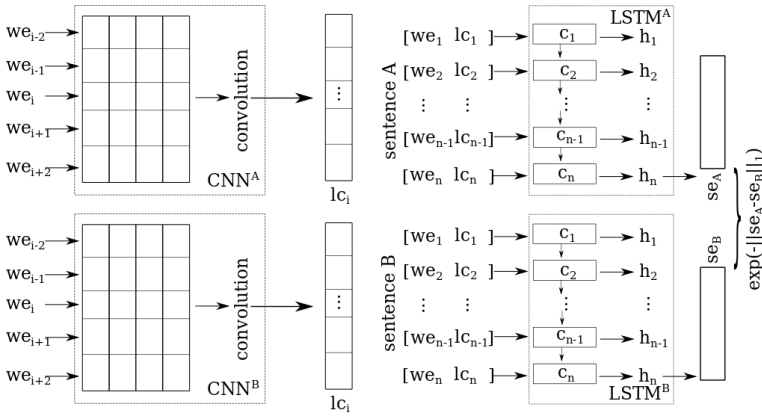


FIGURE 1 – Siamese CNN+LSTM to calculate the similarity of a pair of sentences.

3.1 Neural Network Architecture

Kim trained a simple CNN on top of pre-trained word vectors for the sentence classification task (Kim, 2014). His simple model composed of one layer of convolution achieved excellent results on multiple benchmarks. Inspired by the good results of CNNs in the sentence classification (Kim, 2014), we use a Siamese CNN to generate local contexts for each word in a sentence from its previous and following words. We utilize pre-trained word embeddings¹ to represent these words. Let $we_i \in R^k$ be the k -dimensional word vector corresponding to the i -th word in a sentence. A local context of length l (e.g. $l = 5$) is represented as:

$$\mathbf{x}l_i = \mathbf{x}_{i-2} \oplus \mathbf{x}_{i-1} \oplus \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \mathbf{x}_{i+2} \quad (1)$$

where \oplus is the concatenation operator. Our convolution operation involves a filter $w \in R^{lk}$, which is applied to a window of l words to produce a local context. In more details, our CNN generates the local context of word i by:

$$lc_i = f(\mathbf{w} \cdot \mathbf{x}l_i + \mathbf{b}) \quad (2)$$

where \mathbf{b} is a bias term and f is the hyperbolic tangent function. This filter is connected to every sequence of words in a sentence to deliver a local context for all words.

In order to analyze the general and the local contexts of the word i , we concatenate its pre-trained word embeddings we_i (general semantic and syntactic features that were learned on a large corpus) and its local context lc_i . Our LSTM updates its state c_i and produces an output h_i at time step i in a sentence using the equations described in (Mueller & Thyagarajan, 2016). The last output of our LSTM h_n represents the meaning of a sentence.

Diverse similarity metrics (cosine, Euclidean and Manhattan distances) were tested and we acquired the best outcome with the Manhattan distance $\exp(-||se_A - se_B||) \in [0, 1]$. Since these scores are not optimized for the similarity metric range (1-5), we apply in a post-processing step a regression method using local regression and bandwidth to project our predictions in the correct scale, similarly to (Li & Racine, 2003).

1. Publicly available at: code.google.com/p/word2vec

4 Experimental Setup

We use the SICK dataset to analyze and to test the performance of our system. This dataset contains 9,927 sentence pairs (Marelli *et al.*, 2014) and we split it in 4,927/2,000/3,000 for training/validation/test. Each sentence pair is annotated with a relatedness label $\in [1, 5]$ corresponding to the average relatedness judged by 10 different individuals. The gold scores for relatedness are composed of: 923 pairs within the [1,2) range, 1,373 pairs within the [2,3) range, 3,872 pairs within the [3,4) range, and 3,672 pairs within the [4,5] range.

We initialize our CNN and our LSTM weights with small random Gaussian entries. Our CNN has filters R^{300} and our LSTM has 50-dimensional hidden representations h_t and memory cells c_t . We use a forget bias of 2.5 to model long-range dependencies, Adadelta method to optimize the parameters, and a learning rate of 0.01. We did not identify any improvement with deep LSTMs because of the small amount of data. Like (Mueller & Thyagarajan, 2016), we also augmented our training dataset and we pre-trained our network using the dataset of SemEval 2013 STS task.

5 Results

In order to understand the relevance of the local context for the sentence similarity, we investigated the original Siamese LSTM without local context and compared it with our method using various lengths for the local context: 3, 5, 7, and 9 (Table 1). The original Siamese LSTM analyzes a sentence considering only the general context of words. As expected, the analysis of general and local contexts of words improved the sentence analysis, according to the Pearson’s and Spearman’s correlation coefficients and the Mean Squared Error (MSE) scores. Short or long local contexts did not generate the best results, which shows that short local context (3 words) did not get enough information about the neighborhood of words and long local context (7 words) includes irrelevant information.

Method	r	ρ	MSE
<i>Siamese LSTM (Mueller & Thyagarajan, 2016)</i>	0.8822	0.8345	0.2286
Siamese LSTM (publicly available version) ²	0.8500	0.7860	0.3017
Siamese #local context: 3 + Siamese LSTM	0.8536	0.7909	0.2915
Siamese #local context: 5 + Siamese LSTM	0.8549	0.7933	0.2898
Siamese #local context: 7 + Siamese LSTM	0.8540	0.7922	0.2911
Siamese #local context: 9 + Siamese LSTM	0.8533	0.7890	0.2923
Non-Linear Similarity (Tsubaki <i>et al.</i> , 2016)	0.8480	0.7968	0.2904
Constituency Tree LSTM (Tai <i>et al.</i> , 2015)	0.8582	0.7966	0.2734
Skip-thought+COCO (Kiros <i>et al.</i> 2015)	0.8655	0.7995	0.2561
Dependency Tree LSTM (Tai <i>et al.</i> , 2015)	0.8676	0.8083	0.2532
ConvNets (He <i>et al.</i> , 2015)	0.8686	0.8047	0.2606

TABLE 1 – Pearson (r) and Spearman (ρ) correlation coefficients, and Mean Squared Error for the test set of STS task.

2. We used the public version of Siamese LSTM (Mueller & Thyagarajan, 2016) available at <https://github.com/aditya1503/Siamese-LSTM>, however, we did not get the same results as the ones described in their paper.

The bottom part of Table 1 compares the results of our system and the best state-of-the-art systems. Although our method did not generate the best results, our system is among the top systems and the results were improved with respect to the publicly available version of the original Siamese LSTM.

In order to illustrate how our local context acts on sentence analysis, Table 2 shows at the word level the similarity a pair of paraphrases: *“Her life spanned years of incredible change for women.”* and *“Mary lived through an era of liberating reform for women.”* For each pair of words taken in both sentences, the similarity measured as a cosine distance³ is computed either from general word embeddings (table a) or local contexts of length 5 (table b). The first things to notice is that the two tables have different ranges of values because they each represent a different dimensional space; this means that values must be compared inside each table. Analyzing Table 2a shows that word embeddings preserve general semantic and syntactic relationships of words. In this case, the words are more similar to the words that have similar semantics (1-*“Her”*, 2-*“Mary”* and 2-*“women”*; 1-*“life”* and 2-*“lived”*; 1-*“change”* and 2-*“reform”*) and/or have similar syntactic roles (1-*“of”* and 2-*“for”*). Table 2b highlights that the local context of a word has its semantic and syntactic features based on the words in its window; e.g. the nearest contexts to 1-*“life”* are 2-*“Mary”*, 2-*“lived”*, 2-*“through”* and 2-*“women”* since these local contexts have directly (2-*“lived”*) and indirectly (2-*“Mary”*, 2-*“through”* and 2-*“women”*) similar semantics. This analysis is similar to the syntactic features for the local contexts, e.g. the nearest local context of 1-*“for”* are 2-*“lived”*, 2-*“of”*, 2-*“for”* and 2-*“woman”*. The relevance of local context is strengthened when we analyze phrasal verbs or multi-word expressions in which meaning depends strongly on their previous and their following words.

	Mary	lived	through	an	era	of	liberating	reform	for	women
Her	0.77	0.93	0.90	0.81	1.04	0.92	0.95	0.91	0.80	0.80
life	0.91	0.70	0.89	0.90	0.82	1.00	0.71	0.86	0.88	0.86
spanned	0.88	0.76	0.81	1.01	0.80	0.85	0.92	1.00	0.89	0.93
years	0.88	0.70	0.94	0.88	0.72	0.86	0.92	0.93	0.81	0.86
of	0.93	0.96	0.96	1.09	0.91	0.00	0.99	1.02	0.82	0.91
incredible	0.94	0.89	0.83	0.94	0.84	0.95	0.74	1.04	0.83	0.97
change	0.97	0.90	0.93	0.92	0.85	0.99	0.80	0.67	0.83	0.92
for	0.96	0.97	0.67	0.79	0.89	0.82	0.88	0.92	0.00	0.89
women	0.81	0.96	0.99	0.93	0.92	0.91	0.79	0.88	0.89	0.00

a. Cosine distance between word embeddings.

	Mary	lived	through	an	era	of	liberating	reform	for	women
Her	0.06	0.08	0.09	0.11	0.16	0.12	0.13	0.13	0.09	0.08
life	0.10	0.08	0.09	0.12	0.11	0.13	0.13	0.14	0.10	0.10
spanned	0.15	0.14	0.11	0.11	0.18	0.14	0.14	0.16	0.13	0.12
years	0.13	0.11	0.08	0.13	0.10	0.12	0.11	0.16	0.09	0.09
of	0.12	0.11	0.10	0.12	0.11	0.09	0.12	0.14	0.13	0.11
incredible	0.12	0.12	0.13	0.14	0.19	0.13	0.03	0.16	0.14	0.09
change	0.14	0.13	0.18	0.15	0.18	0.15	0.16	0.02	0.15	0.13
for	0.10	0.09	0.10	0.11	0.12	0.08	0.11	0.12	0.04	0.08
women	0.09	0.07	0.09	0.11	0.11	0.08	0.09	0.14	0.07	0.01

b. Cosine distance between local contexts of length 5.

TABLE 2 – Cosine distance measured between word embeddings (a.) and between the local contexts of length 5 (b.) for each pair of words of two paraphrases.

3. The cosine distance between two vectors u and v is defined by $1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}$.

Table 3 shows four examples of STS scores for multiple levels of similarities. The first pair of sentences describes an example of active and passive voice, with the same meaning (4.9 golden score). The second case is an example of positive and negative sentences (3.3 golden score). The third example is composed of sentences that do not share the same meaning, having 1.0 golden score. Finally, our method helps to determine the semantic relationship of the phrasal verb "wipe off" and the verb "clean" in the last example. Our approach improves the Siamese LSTM analysis by generating better scores. The local context helps to better identify not only similar sentences but also the negation and sentences with different meanings. This local information provides LSTM with a smoother analysis of words and how they connect in a sentence.

Pair of sentences	Golden score	Siamese LSTM	Our approach
<i>Fish is being cooked by a woman. A woman is cooking fish.</i>	4.9	3.84	4.05
<i>The bearded man is not sitting on a train. The bearded man is sitting on a train.</i>	3.3	3.49	3.35
<i>Someone is playing with a toad. The trumpet is being played by a man.</i>	1.0	1.51	1.46
<i>I will wash up if you wipe off the table. I will wash up if you clean the table.</i>	5.0	3.67	4.08

TABLE 3 – Examples of semantic textual similarities using Siamese LSTM and our approach (Siamese #local context: 5 + Siamese LSTM).

To sum up, the local context of words refined the general context analysis. Our approach identified more details about the words and their local as well as general contexts, which usually leads to improved STS scores.

6 Conclusion

STS is an important task for various NLP applications, e.g. Automatic Text Summarization (ATS), Question-Answering, Information Retrieval, etc. Our system combines CNN and LSTM structures to analyze, to identify and to preserve the relevant information in each part of sentences and in the whole sentences. The local context turned out to be useful to get complement information about a word in a sentence and to improve the sentence analysis. In our experiments, the local context improved the prediction of the sentence similarity, by reducing the mean squared error and increasing the correlation scores.

We plan to test other methods to analyze the local context (Ermakova & Mothe, 2016; Zhu *et al.*, 2017). Unfortunately, the dataset we used for the experiments is of a modest size and we did not find larger annotated corpora for this task. Therefore, we also want to lead extrinsic evaluations by measuring how STS acts on ATS systems, depending on whether the original or the modified Siamese LSTM model is used.

Acknowledgments

This work was partially financed by the European Project CHISTERA-AMIS ANR-15-CHR2-0001.

References

- BJERVA J., BOS J., VAN DER GOOT R. & NISSIM M. (2014). The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *SemEval@COLING*, p. 642–646: The Association for Computer Linguistics.
- COLLOBERT R., WESTON J., BOTTOU L., KARLEN M., KAVUKCUOGLU K. & KUKSA P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, **12**, 2493–2537.
- ERMAKOVA L. & MOTHE J. (2016). Query expansion by local context analysis. In *CORIA 2016 - Conférence en Recherche d'Informations et Applications- 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, Toulouse, France, March 9-11, 2016.*, p. 235–250.
- GREFF K., SRIVASTAVA R. K., KOUTNÍK J., STEUNEBRINK B. R. & SCHMIDHUBER J. (2015). LSTM: A search space odyssey. *CoRR*, **abs/1503.04069**.
- HE H., GIMPEL K. & LIN J. J. (2015). Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, p. 1576–1586.
- KIM Y. (2014). Convolutional neural networks for sentence classification. In *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*, p. 1746–1751.
- KIROS R., ZHU Y., SALAKHUTDINOV R., ZEMEL R. S., TORRALBA A., URTASUN R. & FIDLER S. (2015). Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15*, p. 3294–3302, Cambridge, MA, USA: MIT Press.
- LAI A. & HOCKENMAIER J. (2014). Illinois-lh: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, p. 329–334.
- LI Q. & RACINE J. (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis*, **86**(2), 266–292.
- MARELLI M., MENINI S., BARONI M., BENTIVOGLI L., BERNARDI R. & ZAMPARELLI R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, p. 216–223.
- MUELLER J. & THYAGARAJAN A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*, p. 2786–2792: AAAI Press.
- RYCHALSKA B., PAKULSKA K., CHODOROWSKA K., WALCZAK W. & ANDRUSZKIEWICZ P. (2016). Samsung poland nlp team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *SemEval@NAACL-HLT*.
- SEVERYN A., NICOSIA M. & MOSCHITTI R. (2013). Learning semantic textual similarity with structural representations. In *In ACL*.
- TAI K. S., SOCHER R. & MANNING C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, **abs/1503.00075**.

TSUBAKI M., DUH K., SHIMBO M. & MATSUMOTO Y. (2016). Non-linear similarity learning for compositionality. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, p. 2828–2834.

ZHAO J., ZHU T. & LAN M. (2014). ECNU: One stone two birds: Ensemble of heterogeneous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, p. 271–277: Association for Computational Linguistics.

ZHU Q., LI X., CONESA A. & PEREIRA C. (2017). Gram-cnn: a deep learning approach with local context for named entity recognition in biomedical text. *Bioinformatics*.