Siamese Multiplicative LSTM for Semantic Text Similarity

Chao Lv
Science&Technology Center,
Sunshine Insurance Group Co., Ltd
Beijing, China
lyvchao@163.com

Fupo Wang
Science&Technology Center,
Sunshine Insurance Group Co., Ltd
Beijing, China
wfpwork@gmail.com

Jianhui Wang
Science&Technology Center,
Sunshine Insurance Group Co., Ltd
Beijing, China
jianhuimail@gmail.com

Lei Yao
Science&Technology Center,
Sunshine Insurance Group Co., Ltd
Beijing, China
yaoleigle@gmail.com

Xinkai Du*
Science&Technology Center,
Sunshine Insurance Group Co., Ltd
Beijing, China
duxinkai-ghq@sinosig.com

ABSTRACT

Learning the Semantic Textual Similarity (STS) is a critical issue for many NLP tasks such as question answering, document summarization and etc.. In this paper, we combine the Multiplicative LSTM structure with a Siamese architecture which learn to project word embeddings of each sentence into a fixed-dimensional embedding space to represent this sentence. Then these sentence embeddings can be used to evaluate the STS task. We compare with several similar architectures and the proposed method has achieved better results and is competitive with the best state-of-the-art siamese neural network architecture.

CCS CONCEPTS

• **Computing methodologies → Lexical semantics**.

KEYWORDS

siamese network, multiplicative lstm, semantic similarity

ACM Reference Format:
Chao Lv, Fupo Wang, Jianhui Wang, Lei Yao, and Xinkai Du. 2020. Siamese Multiplicative LSTM for Semantic Text Similarity. In *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2020), December 24–26, 2020, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3446132.3446160

1 INTRODUCTION

The task of calculating the similarity between a pair of texts using both direct and indirect relationships between them is defined as measuring Semantic Textual Similarity (STS)[30]. Originally, the STS task is only used for comparing the similarity between short texts such as abstracts and product descriptions[20, 22]. International Workshops on Semantic Evaluation (SemEval) introduced the STS tasks which lead to an increase of the interest that the field received from the research community. And the SemEval tasks also led to the development of the standard STS datasets like the SICK corpus[21] and the standardized the similarity score as a numerical value between 1 and 5[12].

Semantic Text Similarity is a fundamental and important task in Natural Language Processing (NLP) which can be greatly improved by modeling the underlying semantic representations of the compared sentences. STS can be applied in many text related research such as information retrieval, text classification, semantic search, question answering, sentiment analysis, machine translation and the others[13, 14].

Gomaa and Fahmy [14] categorized the text similarity measuring method into three kinds: string-based, corpus-based and knowledge-based. While Mamdouh Farouk [13] classified the approaches of calculating sentences similarity based on the adopted methodology into three categories: word-to-word based, structure-based, and vector-based. Beyond the corpus-based and knowledge-based method, the categorized method proposed by Dhivya Chandrasekaran and Vijay Mago[8] also include the deep neural network-based methods and hybrid methods.

Numerous STS measuring methods have been proposed by researchers as a result of SemEval workshops for the growing importance of having a better STS metric [29]. With the development of amount annotated data, many deep learning methods can be introduced to solve the STS tasks. Shuang Peng et al. [27] categorized recently proposed scheme into three kinds: the siamese network, matching-aggregation and BERT-based method. In this article, we focus on the siamese architecture and all the various variations.

Siamese networks are suitable for finding similarity or a relationship between two comparable things[29]. Siamese network was first introduced in [6] to verify signatures written on a pen-input tablet. Chopra et al. [10] used this method to learn a similarity for the face verification. Koch etc.[18] also use this network to check the similarity between images. Mueller [24] and Neculoiu [25] first use the siamese architecture to measure sentence similarity by using two LSTMs to construct the comparing structure with Manhattan distance metric. Ichida [16] proposed the Siamese GRU structure to improve Mueller's work [24].

Siamese architectures are good at measuring STS, for the case that we can chose the same kind of inputs which are able to be

*Corresponding Author

ACAI 2020, December 24–26, 2020, Sanya, China

ACM ISBN 978-1-4503-8811-5/20/12…$15.00
https://doi.org/10.1145/3446132.3446160

processed by a similar model. In this way the networks will represent semantics by using vectors, making them easier to compare pairs of sentences. And the weights are shared across sub networks, so there are fewer parameters to train. This means that the siamese network require less training data and less tendency to over-fit.

Krause2016 etc. [19] proposed a hybrid architecture named multiplicative LSTM (mLSTM). The mLSTM is composed of the factorized hidden-to-hidden transition of mRNNs and the gating framework from LSTMs. The authors put the mRNN's intermediate state m_t into each gating units of the LSTM to combine the mRNN and LSTM architectures.

By the multiplicative LSTM architecture, so in this article, we construct the Siamese Multiplicative LSTM network by using the siamese network architecture and the mLSTM to evaluate STS task which get a improved achievement.

This paper is organized as follows: in Section 2, we make an overview of relevant works for STS evaluation. The proposed Siamese Multiplicative LSTM network is introduced in Section 3. Then we show the evaluation result on the several STS datasets comparing with the other two schemes in Section 4. Finally, we give our conclusion in Section 5.

2 RELATED WORKS

Mueller and Thyagarajan [24] proposed a Siamese LSTM structure to represent the sentences which is encoded using the pre-trained word embedding vectors. And in this structure, the same weights are chosen to encode sentences which produce comparable sentence representations for similar sentence pairs. Then the closeness scores can be predicted by use of the Manhattan distance between the pair sentence representations.

Neculoiu et al. [25] proposed a character level two layer bidirectional LSTM network to evaluate the similarity between job titles by using the contrastive loss function. Semantic differences can be captured by this Siamese network while being invariant under non-semantic string differences [26].

Kenter et al. [17] presented Siamese CBOW model which trained the embeddings of words in a sentences directly for the purpose of being averaged. And this model use the cosine similarity between the two sentence vectors as the final semantic similarity score.

Yin et al. [31] proposed the attention based CNN model for considering the mutual influence between the sentence pair which is different from learning the representation of sentence separately and got better performance.

In [26], the proposed hybrid siamese network architectures which had the following characters: (i) the encoders shared parameters; (ii) the comparison between two sentences was measured by euclidean distance; (iii) multi-layer perceptron was added to optimize the contrastive and the logistic loss. By the above traits, the authors claimed that this model can exploit a more informative feedback. The first deep learning model is the sentence encoder. A sentence of length n is able to be expressed as a sequence of words $(w_1, \ldots, w_n)$, which are drawn from a vocabulary V. And each word is represented as a vector, $\mathbf{w} \in \mathbb{R}^d$, looked up into an embedding matrix, $\mathbf{E} \in \mathbb{R}^{d\times|V|}$. Then the sentence encoder f takes a sequence of words as in input, embeds and transforms them into a fixed-sized vector. The function f is used to encode both sentences in a pair, by sharing the same set of weights. "This is a typical of a siamese setting, where the same network maps the two objects of a pair into a low dimensional space, where their distance is small if they are similar." At last, the euclidean distance of two sentences s_1 and s_2 will be calculated as follows:

$$d(s_1, s_2) = \sqrt{\sum_{i=1}^{n} \left(f(s_1)_i - f(s_2)_i\right)^2}$$

The output of the siamese encoder is fed into a multilayer perceptron (MLP) which is the architecture of the proposed hybrid siamese network. The MLP takes the concatenation of the sentence representations and their distance $c = [f(s_1); f(s_2); d(s_1, s_2)]$, in input, and outputs the probability of a match between the two sentences. By the above traits, the authors claimed that this model can exploit a more informative feedback.

In [16], Ichida et al. proposed a Siamese neural network architecture combined with two GRUs (Gating Recurrent Units) which have fewer parameters than LSTM units. A GRU cell consists of two gates that controls flow data through states. Gate r_t controls updates on internal memory, which is not propagated to next state. Gate z_t controls how much of internal memory should be considered on next state. The following equations represent the operations realized by gates r_t and z_t in order to results, and show how next hidden state is computed in a GRU unit. This scheme also used the Manhattan distance function to calculate the difference between encoded sentences. The authors compared the two recurrent neural network architectures LSTM and GRU based on the SICK dataset.

$$\begin{aligned} r_t &= \sigma(W_r h_{t-1} + U_r x_t + b_r) \\ z_t &= \sigma(W_z h_{t-1} + U_z x_t + b_z) \\ h_t &= z_t \otimes h_{t-1} \oplus (1 - z) \otimes \tanh(W_h x_t + U_h (r_t \otimes h_{t-1}) + b_h)) \end{aligned}$$

In [28], the authors used a siamese CNN structure to represent the relevance of words among the sentences. In this scheme, the authors uses a convolution network to take account of the local context of words and an LSTM to consider the global context of sentences. Then a siamese LSTM was used to analyze the whole sentence based on its words and the local context. And also the Manhattan distance was used to predict the semantic similarity between the pair of sentences.

Peng et al.[27] presented an enhanced recurrent convolutional neural network (Enhanced-RCNN) model to evaluate the sentence similarity. By this architecture, the siamese multi-layer CNNs were used to extract key information from the sentence pair and the attention-based RNNs were utilized to capture the interactive effects between the two sentences. In details, the proposed Enhanced-RCNN model is composed of three components: input encoding, interactive sentence representation, and similarity modeling. There are three components in the input encoding section, such as bidirectional GRU (BiGRU)[9], CNN, and Pooling. The authors use BiGRU to extract the sequence and context features of sentences and to memorize the long-distance information. CNN is good at capturing "keywords and phrases information" in sentences comparing with BiGRU. This characteristic has been used in summarization research [7]. And the pooling layer in this model processes the CNN output and can help reduce the model parameters and enhance the extensiveness and robustness. However, the module of interactive sentence representation aimed to obtain an appropriate representation for each sentence with consideration of the interactive effects from the other sentence. With the interactive sentence representation derived, the authors designed a particular fusion layer to combine the two sentence representation separately for overall similarity modeling.

3 SIAMESE MULTIPLICATIVE LSTM NETWORK

3.1 Siamese Multiplicative LSTM Model

The proposed Siamese Multiplicative LSTM Architecture is shown in Fig.1. We construct two networks MuLSTM^A and MuLSTM^B to present the sentence pair separately. However in this place, we only focus on the siamese architecture with tied weights. So $\text{MuLSTM}^A = \text{MuLSTM}^B$ in this network. Each sentence will be encoded to word vectors before they are sent into the Multiplicative LSTM network. And we choose GoogleNews-vectors [23] to represent words of sentence. Then these word embeddings of each sentence can be constructed as one fixed-dimensional embedding to represent this sentence.

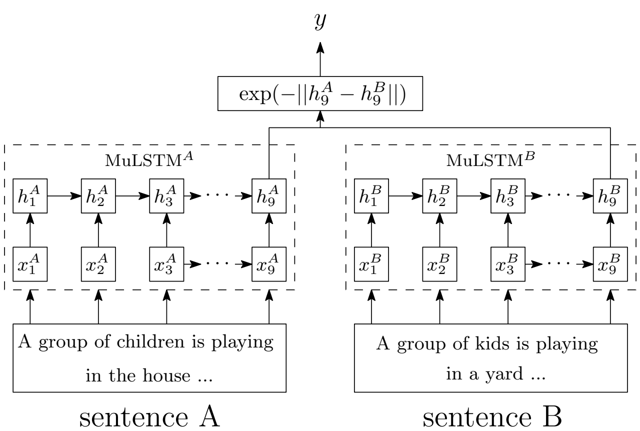

Figure 1: Siamese Multiplicative LSTM Architecture

3.2 Multiplicative LSTM

The classic LSTM is commonly used RNN architecture which uses a series of multiplicative gates to control the information flows in and out of internal states of the network[15]. The variants of LSTM used by Multiplicative LSTM is shown as follows.

The input layer x_t and the previous hidden state h_{t-1} outputs to the current hidden state.

$$\hat{h}_t = W_{hx}x_t + W_{hh}h_{t-1}$$

There are also three gating units, input gate i, output gate o, and forget gate f which have both recurrent and feed-forward connections:

$$i_t = \sigma\left(W_{ix}x_t + W_{ih}h_{t-1}\right)$$
$$o_t = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1}\right)$$
$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1}\right),$$

where σ is the logistic function. The internal state vector c_t is written from each hidden unit which is controlled by the input gate. And the forget gate determines how much of the previous internal state c_{t-1} is preserved. This combination of write and forget gates allows the network to control what information should be stored and overwritten across each time-step. The internal state is updated by

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(\hat{h}_t\right).$$

The output gate also controls how much of each unit's activation is preserved. So the LSTM cell is able to keep information that is not relevant to the current output, while may be relevant later. The final output of the hidden state can be calculated as

$$h_t = \tanh(c_t) \odot o_t.$$

According to the variants of classic LSTM described above, the Multiplicative LSTM architecture [19] is composed of the flexible input-dependent transitions of mRNNs, the long time lag and the information control of LSTMs. The complex transitions result from the factorized hidden weight matrix would be controlled easier for the case of LSTMs' gated units. We can use utilize more flexible input-dependent transition functions than in regular mRNNs for adding the additional sigmoid input and forget gates featured in LSTM units. Also in this structure, m_t would be shared across all LSTM unit types. And the mLSTM can be described as follows.

$$m_t = (W_{mx}x_t) \odot (W_{mh}h_{t-1})$$
$$\hat{h}_t = W_{hx}x_t + W_{hm}m_t$$
$$i_t = \sigma\left(W_{ix}x_t + W_{im}m_t\right)$$
$$o_t = \sigma\left(W_{ox}x_t + W_{om}m_t\right)$$
$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_t\right)$$

4 EXPERIMENTS

4.1 Data

The SICK dataset [21] consists of 10,000 English sentence pairs. And each pair is annotated with a relatedness continuous score $\in [1, 5]$ corresponding to the average relatedness judged by 10 different individuals.

The STS dataset consists of several datasets which were built by the organizers of the SemEval shared task. And in this article, we

Table 1: Datasets for Semantic Similarity Used

Dataset Name	Sentence Pairs	Similarity Score	year
SICK [21]	10,000	[1, 5]	2014
STS2012 [4]	5250	[0, 5]	2012
STS2013 [5]	2250	[0, 5]	2013
STS2014 [2]	3750	[0, 5]	2014
STS2015 [1]	3000	[0, 5]	2015
STS2016 [3]	1186	[0, 5]	2016
MSRP [11]	5801	0, 1	2004
QQP	404,290	0, 1	2017

Table 2: Results of Evaluated Models

Dataset Name	MaLSTM			MaGRU			MuLSTM		
	r	ρ	MSE	r	ρ	MSE	r	ρ	MSE
SICK	0.6131	0.4906	0.7526	0.6561	0.5326	0.6616	**0.7374**	**0.6024**	**0.5093**
STS2012	0.5264	0.5483	1.8570	0.7554	0.7128	**0.9645**	**0.7626**	**0.7469**	1.0707
STS2013	0.2683	0.2752	3.0494	**0.3968**	**0.4038**	2.6756	0.3218	0.3050	**2.3548**
STS2014	0.1355	0.1325	3.4446	0.3995	**0.4072**	2.2936	**0.4068**	0.3841	**2.1689**
STS2015	0.4753	0.4734	2.0249	0.5154	0.5151	1.8389	**0.5810**	**0.5682**	**1.5945**
STS2016	0.4956	0.4945	2.2235	**0.5701**	**0.5702**	**1.9946**	0.4966	0.4923	2.1331
MSRP	0.5758	0.5578	0.1669	0.6427	0.6091	0.1376	**0.6594**	**0.6322**	**0.1332**
QQP	0.0837	0.1119	0.3255	0.1230	0.1219	**0.3081**	**0.1606**	**0.1564**	0.3094

will use STS 2012 ∼ 2016. Each sentence pair is annotated with the similarity score in the scale of [0, 5].

The Microsoft Research paraphrase corpus (MSRP) has 5801 pairs of sentences. And each pair of sentence has a "Quality" tag with a binary value (0 or 1) to indicate whether human raters' judgement of the pair sentences to be similar enough.

The Quora Question Pairs (QQP) [1] dataset consists of a training set of 404,290 question pairs and a test set of 2,345,795 question pairs which is provided as part of a Kaggle competition.

All the details of the above mentioned dataset used in this paper are shown in Table 1.

4.2 Experiments Settings

Our LSTM use 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 parameters, and a learning rate of 0.001. To represent each sentence with the same length for the convenience of comparison, we will pad the short sentence with the length 25. And the longer ones will be cut to 25. For the training and test data sets, most lengths of the experiment sentences will be less than 25.

4.3 Evaluation

Table 2 show the results of the proposed model comparing with other architectures on the several dataset. In this table, we show the best result of three models as MaLSTM[24], MaGRU [16] and the proposed method named MuLSTM. MaLSTM is the baseline that we defined for this research. All the models are measured by the three evaluation metrics usually employed in the STS task: Pearson correlation (r), Spearman correlation (ρ) and Mean Square Error (MSE). As shown in Table 2, for example the SICK test, the r value of our proposed scheme get 0.7374 which is the best comparing to the other two methods: 0.6131, 0.6561. And $\rho = 0.6024$, $MSE = 0.5093$ is also the best result for the STS dataset. For the evaluation metric r, our method has got the best reuslt on the SICK, STS2012, STS2014, STS2015, MSRP and QQP datasets. For the evaluation metric ρ, the proposed method has got the best result on the SICK, STS2012, STS2015, MSRP and QQP datasets. And for the MSE, the MuLSTM has got the best result on the SICK, STS2013, STS2014, STS2015 and MSRP datasets. By the result, our proposed scheme most get the best value on r, ρ and MSE.

[1]https://www.kaggle.com/c/quora-question-pairs

5 CONCLUSION

In this paper, we compare the proposed Siamese Multiplicative LSTM network with the other two common Siamese neural network architectures for STS evaluation: MaLSTM and MaGRU. The experiment shows that the proposed scheme has a better result on the three metrics: Pearson correlation, Spearman correlation and Mean Square Error. We also choose several padding parameters such as 20, 25 and 30. Then we find that this length also have a great influence on the measuring results.

REFERENCES

[1] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, 252–263.

[2] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, Dublin, Ireland, 81–91.

[3] 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 *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, 497–511.

[4] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Association for Computational Linguistics, Montréal, Canada, 385–393.

[5] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, Atlanta, Georgia, USA, 32–43.

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems* (Denver, Colorado) *(NIPS'93)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 737–744.

[7] Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang. 2015. Learning Summary Prior Representation for Extractive Summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, 829–833. https://doi.org/10.3115/v1/P15-2136

[8] Dhivya Chandrasekaran and Vijay Mago. 2020. Evolution of Semantic Similarity – A Survey. arXiv:2004.13820 [cs.CL]

[9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[10] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1.

[11] Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. COLING, Geneva, Switzerland, 350–356.

[12] Claire Cardie Daniel Cer Mona Diab Eneko Agirre, Carmen Banea. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. (2014).

[13] Mamdouh Farouk. 2019. Measuring Sentences Similarity: A Survey. *Indian Journal of Science and Technology* 12, 25 (Jul 2019), 1–11.

[14] Wael H. Gomaa and Aly A. Fahmy. 2013. A Survey of Text Similarity Approaches. *International Journal of Computer Applications* 68, 13 (April 2013), 13–18. Full text available.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[16] A. Y. Ichida, F. Meneguzzi, and D. D. Ruiz. 2018. Measuring Semantic Similarity Between Sentences Using A Siamese Neural Network. In *2018 International Joint Conference on Neural Networks (IJCNN)*. 1–7.

[17] Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 941–951.

[18] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese Neural Networks for One-shot Image Recognition.

[19] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative LSTM for sequence modelling. *CoRR* abs/1609.07959 (2016). arXiv:1609.07959

[20] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering* 18, 8 (2006), 1138–1150.

[21] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 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'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 216–223.

[22] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1* (Boston, Massachusetts) *(AAAI'06)*. AAAI Press, 775–780.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. GoogleNews-vectors-negative300.bin.gz - Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013). https://code.google.com/archive/p/word2vec/

[24] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2786–2792.

[25] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Berlin, Germany, 148–157.

[26] Massimo Nicosia and Alessandro Moschitti. 2017. Accurate Sentence Matching with Hybrid Siamese Networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 2235–2238.

[27] Shuang Peng, Hengbin Cui, Niantao Xie, Sujian Li, Jiaxing Zhang, and Xiaolong Li. 2020. Enhanced-RCNN: An Efficient Method for Learning Sentence Similarity. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2500–2506. https://doi.org/10.1145/3366423.3379998

[28] Elvys Linhares Pontes, Stéphane Huet, Andréa Carneiro Linhares, and Juan-Manuel Torres-Moreno. 2018. Predicting the Semantic Textual Similarity with Siamese CNN and LSTM. *CoRR* abs/1810.10641 (2018). arXiv:1810.10641

[29] Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2019. Semantic Textual Similarity with Siamese Neural Networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. INCOMA Ltd., Varna, Bulgaria, 1004–1011.

[30] Vasile Rus, Mihai Lintean, Rajendra Banjade, Nobal Niraula, and Dan Stefanescu. 2013. SEMILAR: The Semantic Similarity Toolkit.

[31] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.