Semantic Textual Similarity with Siamese Neural Networks

Tharindu Ranasinghe, Constantin Orăsan and Ruslan Mitkov
Research Group in Computational Linguistics
University of Wolverhampton, UK
{t.d.ranasinghehettiarachchige, c.orasan, r.mitkov }@wlv.ac.uk

Abstract

Calculating the Semantic Textual Similarity (STS) is an important research area in natural language processing which plays a significant role in many applications such as question answering, document summarisation, information retrieval and information extraction. This paper evaluates Siamese recurrent architectures, a special type of neural networks, which are used here to measure STS. Several variants of the architecture are compared with existing methods.

1 Introduction

Measuring Semantic Textual Similarity (STS) is the task of calculating the similarity between a pair of texts using both direct and indirect relationships between them (Rus et al., 2013). Originally, the work on STS largely focused on similarity between short texts such as abstracts and product descriptions (Li et al., 2006; Mihalcea et al., 2006). The introduction of the STS tasks at the International Workshops on Semantic Evaluation (SemEval) lead to an increase of the interest that the field received from the research community. The SemEval tasks also led to the development of standard datasets like the SICK corpus (Bentivogli et al., 2016) and standardised the similarity score as a numerical value between 1 and 5 (Agirre et al., 2014).

Having a good STS metric is very important in many natural language processing (NLP) applications. As an example, for certain types of question answering systems, having an accurate STS component is the key to success since the questions with similar meanings can be answered similarly (Majumder et al., 2016). STS is also important in translation memories retrieval and matching (Gupta et al., 2014b). Translation memories help translators by finding in the database they maintain previously translated sentences, which are similar to the one to be translated, and retrieving their translations. Hence, accurate STS methods are beneficial for translation memory.

Given the growing importance of having a good STS metric and as a result of the SemEval workshops, researchers have proposed numerous STS methods. Most of the early approaches were based on traditional machine learning and involved heavy feature engineering (Béchara et al., 2015). With the advances of word embeddings, and as a result of the success neural networks have achieved in other fields, most of the methods proposed in recent years rely on neural architectures (Tai et al., 2015; Shao, 2017). Neural networks are preferred over traditional machine learning models as they generally tend to perform better than traditional machine learning models. They also do not rely on linguistic features which means they can be easily applied to languages other than English. The architecture employed in this paper is a special class of neural networks called *Siamese neural networks*. These networks contain two or more identical sub-networks. The networks are identical in the sense that they have the same configuration with the same parameters and weights. In addition, parameter updating is mirrored across these sub-networks.

Siamese networks are popular among tasks that involve finding similarity or a relationship between two comparable things. They have been proven successful in tasks like signature verification (Bromley et al., 1993), face verification (Chopra et al., 2005), image similarity (Koch et al., 2015) and have been re-

Proceedings of Recent Advances in Natural Language Processing, pages 1004–1011,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_116

cently used successfully in sentence similarity (Neculoiu et al., 2016). Siamese architectures are good in these tasks because, when the inputs are of the same kind, it makes sense to use a similar model to process similar inputs. In this way the networks will have representation vectors with the same semantics, making them easier to compare pairs of sentences. Given that the weights are shared across sub networks there are fewer parameters to train, which in turn means they require less training data and less tendency to over-fit. Given the amount of human labour required to produce datasets for STS, Siamese neural networks can prove the ideal solution for the STS task.

This paper explores the performance of several architectures which use Siamese neural networks for STS. The rest of the paper is organized as follows. Section 2 briefly describes several approaches used to measure sentence similarity focusing more on Siamese neural networks. Section 3 contains information about the settings of the experiments carried out in this paper including the datasets employed here and the different architectures explored. The architectures are evaluated in Section 4. The paper finishes with conclusions.

2 Related Work

Given that a good STS metric is required for a variety of NLP fields, researchers have proposed a large number of such metrics. Before the shift of interest in neural networks, most of the proposed methods relied heavily on feature engineering. A typical example is (Gupta et al., 2014a) which employed 20 linguistic features fed into a support vector machine regressor. The top system (Zhao et al., 2014) in task 1 in SemEval 2014 has used seven types of features including text difference measures, common text similarity measures etc. (Zhao et al., 2014). Then they have fed it in to several learning algorithms like support vector machine regressor, Random Forest, Gradient boosting etc (Zhao et al., 2014). With the introduction of word embedding models, researchers focused more on neural representation for this task. Many of the leading teams in the STS task at Semeval 2017 used some kind of neural network architecture which employed word embeddings (Shao, 2017). As an example, Maharjan et al. (2017) used an ensemble of traditional machine learning models and deep learning models in their top performing system at Semeval 2017 STS task.

There are two main approaches which employ neural representation models: supervised and unsupervised. Unsupervised approaches use pretrained word/sentence embeddings directly for the similarity task without training a neural network model on them. Such approaches have used cosine similarity on sent2vec (Pagliardini et al., 2018), InferSent (Conneau et al., 2017), Doc2Vec (Le and Mikolov, 2014) and smooth inverse frequency with GloVe vectors (Arora et al., 2017).

Supervised approaches use neural networks to project word embeddings to fixed dimensional vectors which are trained to capture the semantic meaning of the sentence. Recently, many neural network architectures have been used to calculate sentence similarity. He et al. (2015) propose an elaborate convolutional network (ConvNet) variant which infers sentence similarity by integrating various differences across many convolutions at varying scales.

Kiros et al. (2015) propose the skip-thoughts model, which extends the skip-gram approach of word2vec from the word to sentence level. This model feeds each sentence into an Recurrent Neural Network (RNN) encoder-decoder with Gated Recurrent Unit (GRU) activations. They attempt to reconstruct the immediately preceding and following sentences. For the sentence similarity task, they obtain skip-thought vectors for sentence pairs. Then a separate classifier is trained using features derived from differences and products between skip-thought vectors for each pair of sentences.

Tai et al. (2015) propose Tree-LSTMs (Long short-term memory) which generalize the order-sensitive chain-structure of standard LSTMs to tree-structured network topologies. Each sentence is first converted into a parse tree using a separately trained parser, and the Tree-LSTM composes its hidden state at a given tree node from the corresponding word as well as the hidden states of all child nodes. The hope is that by reflecting syntactic properties of a sentence, the parse tree structured network can propagate necessary information more efficiently than a sequen-

tially restricted architecture. The output from Tree-LSTM can be used for sentence similarity task as the same way as Kiros et al. (2015), where representations of the input sentences are now produced by Tree-LSTMs rather than skip-thoughts.

Our proposed model also represents sentences using neural networks whose inputs are word vectors learned separately from a large corpus. But unlike the models proposed by Kiros et al. (2015) and Tai et al. (2015) the sole target of our objective function is to calculate sentence similarity. In order to have an objective function that solely focus on similarity we need an architecture which is capable of handling two sentences parallelly. To do that we use a special kind of neural network architecture: Siamese neural network architecture.

Siamese recurrent neural networks have been recently used in STS tasks. The MALSTM architecture (Mueller and Thyagarajan, 2016) uses two identical LSTM networks trying to project zero padded word embeddings of a sentence to fixed sized 50 dimensional vectors using Manhattan distance as the similarity function between 2 sub networks. Mueller and Thyagarajan (2016) report that it performs better than other neural network models like Tree-LSTM (Tai et al., 2015). This inspired us to use this model and extend it. This research proposes new variants of the MALSTM architecture for predicting STS [1].

3 Settings of the Experiments

3.1 Data Sets

The experiments presented in this paper were carried out using the SICK dataset (Bentivogli et al., 2016) and SemEval 2017 Task 1 dataset (Cer et al., 2017) which we will refer as STS2017 dataset.

The SICK data contains 9927 sentence pairs with a 5,000/4,927 training/test split which were employed in the SemEval tasks. Each pair is annotated with a relatedness score between [1,5] corresponding to the average relatedness judged by 10 different individuals. In order to generate more training data we used thesaurus-based augmentation (Miller, 1992) and added 10,022 additional training examples. Evaluation was done with the SICK test data. Mueller and Thyagarajan (2016) uses the same thesaurus-based data augmentation in their research.

The STS2017 test datset had 250 sentence pairs annotated with a relatedness score between [1,5]. As the training data for the competition, participants were encouraged to make use of all existing data sets from prior STS evaluations including all previously released trial, training and evaluation data [2]. Once we combined all datasets from prior STS tasks we had 8277 sentence pairs for training.

3.2 Proposed Architectures

The basic structure of the Siamese neural network architecture used in our experiments is shown in Figure 1. It consists of an embedding layer which represents each sentence as a sequence of word vectors. This sequence of word vectors is fed into a Recurrent Neural Network (RNN) cell which learns a mapping from the space of variable length sequences of 300-dimensional vectors into a 50 dimensional vector. The sole error signal backpropagated during training, stems from the similarity between these 50 dimensional vectors, which can be also used as a sentence representation. Initially, the similarity function we used was based on Manhattan distance. To make sure that the prediction is between 0 and 1, we took the exponent of the negative Manhattan distance between 2 sentence representations. The similarity function was adopted from Mueller and Thyagarajan (2016). The proposed variants of our architecture are:

1. LSTM - Block A in Figure 1 contains a single LSTM cell. This is the architecture suggested by Mueller and Thyagarajan (2016)

2. Bi-directional LSTM - Block A in Figure 1 contains a single Bi-directional LSTM cell. Bi-directional LSTM tends to understand the context better than Unidirectional LSTM (Schuster and Paliwal, 1997).

3. GRU - Block A in Figure 1 contains a single GRU cell. GRUs have been shown

[1]The code is available on "https://github.com/TharinduDR/Siamese-Recurrent-Architectures"

[2]http://alt.qcri.org/semeval2017/task1/

Approach	τ	ρ	MSE
Illinois-LH (Lai and Hockenmaier, 2014)	0.7993	0.7538	0.3692
UNAL-NLP (Jiménez et al., 2014)	0.8070	0.7489	0.3550
Meaning Factory (Bjerva et al., 2014)	0.8268	0.7721	0.3224
ECNU (Zhao et al., 2014)	0.8414	NA	NA
Skip-thought+COCO (Kiros et al., 2015)	0.8655	0.7995	0.2561
Dependency Tree-LSTM (Tai et al., 2015)	0.8676	0.8083	0.2532
ConvNet (He et al., 2015)	0.8686	0.8047	0.2606
Bi-directional LSTM†	0.8743	0.8251	0.2391
GRU + Capsule + Flatten†	0.8786	0.8286	0.2301
MALSTM (Baseline) (Mueller and Thyagarajan, 2016)	0.8822	0.8345	0.2286
LSTM: Adagrad†	0.8831	0.8364	0.2195
GRU + Attention†	0.8843	0.8372	0.2163
LSTM + Attention†	0.8886	0.8386	0.2142
Bi-directional GRU†	0.8896	0.8390	0.2125
GRU†	0.8901	0.8396	0.2112

Table 1: Pearson correlation (τ), Spearman correlation (ρ), and Mean Square Error (MSE) for the SICK test set.

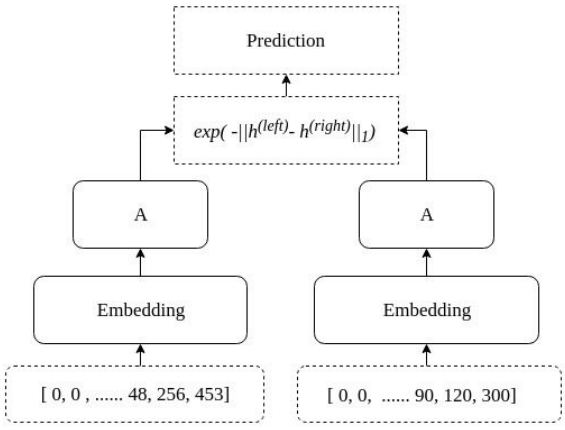

Figure 1: Basic structure of the Siamese neural network. Unit A is changed over the architectures.

to exhibit better performance on smaller datasets (Chung et al., 2014).

4. Bi-directional GRU - Block A in Figure 1 contains a single Bi-directional GRU cell. Bi-directional GRUs tend to understand the context better than Unidirectional GRUs (Vukotic et al., 2016).

5. LSTM + Attention - Block A in Figure 1 contains a single LSTM cell with self attention (Bahdanau et al., 2014).

6. GRU + Attention - Block A in Figure 1 contains a single GRU cell with self attention (Bahdanau et al., 2014).

7. GRU + Capsule + Flatten - Block A in Figure 1 contains a GRU followed by a capsule layer and a flatten layer. Dynamic routing used between capsules performs better than a traditional max-pooling layer (Sabour et al., 2017).

4 Evaluation Results

Table 1 shows the results obtained for the proposed architectures on the SICK test set. The table only reports the best results for each ar-

chitecture [3]. The first group of results are top SemEval 2014 submissions and the second group are recent neural network methods (best result from each paper shown). † denotes the experiments we conducted in this research. All the models were evaluated using the three evaluation metrics normally employed in the STS tasks: Mean Square Error (MSE), Pearson correlation (τ) and Spearman correlation (ρ).

MALSTM (Mueller and Thyagarajan, 2016) is the baseline that we defined for this research. The baseline is the best result achieved by the architecture reported in Mueller and Thyagarajan (2016). Interestingly, we were able to beat the baseline using the same architecture using the Adagrad optimiser (Duchi et al., 2011) (LSTM:Adagrad).

The best result was obtained when block A in figure 1 contains a single GRU. As can be seen in the table 1, the proposed architecture outperformed both the benchmark and all the other architectures in all 3 evaluation metrics.

As can be seen in Table 1, the architectures with bidirectional GRU, LSTM with Attention and GRU with Attention also surpassed the benchmark. However uni-directional GRU out performed them on all 3 evaluation metrics.

We experimented with other similarity functions and other embedding models. Using Euclidean distance for the similarity function instead of Manhattan distance did not improve the results for the model because semantically different sentences could end up being represented by nearly identical vectors due to the vanishing gradients of the Euclidean distance (Chopra et al., 2005). Changing the embedding model to GloVe (Pennington et al., 2014), fastText (Mikolov et al., 2018) or concatenating them with word2vec model did not improve the results either. For this reason, none of these results are presented here.

The Siamese neural network with GRU was tested with a cyclical learning rate (Smith, 2015), which has the advantage of forcing the model to find another local minimum if the current minimum is not robust and makes the model generalize better to unseen data. However, neither cyclical learning rate nor reducing learning rate on plateau increased the performance further. We do not report these results too.

[3]These results are reported in Marelli et al. (2014)

Approach	τ
FCICU (Hassan et al., 2017)	0.8280
BIT (Wu et al., 2017)	0.8400
ECNU (Tian et al., 2017)	0.8518
DT Team (Maharjan et al., 2017)	0.8536
Bi-directional LSTM†	0.8540
GRU + Capsule + Flatten†	0.8545
RTV	0.8547
MALSTM	0.8651
LSTM: Adagrad†	0.8692
GRU + Attention†	0.8725
LSTM + Attention†	0.8743
Bi-directional GRU†	0.8750
GRU†	0.8792

Table 2: Pearson correlation (τ) for STS2017 test set.

Table 2 shows the results obtained for STS2017 test dataset comparing our experiments with other top performing models in SemEval 2017 Task 1 (Cer et al., 2017). † denotes the experiments we conducted in this research. As SemEval 2017 Task 1 used Pearson correlation (τ) to evaluate the submissions, we evaluated our models using Pearson correlation (τ) too.

GRU based Siamese neural network performs better than existing systems for STS2017 dataset too, as it is shown in table 2.

4.1 Error Analysis

In order to understand better why the GRU based architecture performed better than the LSTM baseline, we compared sentences where the GRU architecture was better. Table 3 shows examples of such sentences from the SICK testset. Our analysis suggests that the GRU based architecture handles the additional words better than LSTM. Mueller and Thyagarajan (2016) report that their architecture does not perform well with active-passive equivalence. However, as shown in Table 4,

Sentence 1	Sentence 2	GOLD	LSTM	GRU
The people are walking on the road beside a beautiful waterfall	The people are walking on the road beside a waterfall, which is beautiful	0.9750	0.5260	0.9569
The woman is frying a chop of breaded pork	The woman is frying a breaded pork chop	0.9250	0.5278	0.8561
A white dog is standing on the leaves on the ground	A dog, which is white, is standing on fallen leaves	0.9500	0.3611	0.7618
The man is erasing the other man's work from the board	The man is erasing the drawing on the board	0.7500	0.5128	0.7584

Table 3: Example sentence pairs from the SICK test data. LSTM denotes the baseline and GRU the best model

Sentence 1	Sentence 2	GOLD	LSTM	GRU
A man is mixing vegetables in a pot	Vegetables are being mixed in a pot by a man	0.9750	0.6684	0.8154
Carrots are being sliced by a woman	A woman is slicing carrots	1.0000	0.6739	0.7206
The elephant is being ridden by the woman	The woman is riding the elephant	0.9500	0.2249	0.5939

Table 4: Example active-passive sentence pairs from the SICK test data.

our architecture performs slightly better than the LSTM based architecture.

5 Conclusions

This paper evaluated several neural architectures based on Siamese recurrent neural network for calculating semantic similarity between pairs of texts. Most of these architectures fared better than the approach proposed in (Mueller and Thyagarajan, 2016). The variant with a GRU performed best, capitalising on GRU's ability to exhibit better performance on smaller datasets like the ones available for STS. Our architectures can be easily ported to other languages which have training data available, and we are currently experimenting with other languages.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *SemEval@COLING*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR 2017*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. 2015. Miniexperts: An svm approach for measuring semantic textual similarity. In *SemEval@NAACL-HLT*.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50:95–124.

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *SemEval@COLING*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a siamese time delay neural network. *IJPRAI*, 7:669–688.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017.

Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval@ACL*.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:539–546 vol. 1.

Junyoung Chung, ÃĞaglar GülÃğehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Rohit Gupta, Hanna Béchara, Ismaïl El Maarouf, and Constantin Orasan. 2014a. Uow: Nlp techniques developed at the university of wolverhampton for semantic similarity and textual entailment. In *SemEval@COLING*.

Rohit Gupta, Hanna Bechara, and Constantin Orasan. 2014b. Intelligent translation memory matching and retrieval metric exploiting linguistic technology. *Proc. of Translating and the Computer*, 36:86–89.

Basma Hassan, Samir E. AbdelRahman, Reem Bahgat, and Ibrahim Farag. 2017. Fcicu at semeval-2017 task 1: Sense-based language independent semantic textual similarity approach. In *SemEval@ACL*.

Hua He, Kevin Gimpel, and Jimmy J. Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*.

Sergio Jiménez, George Dueñas, Julia Baquero, and Alexander F. Gelbukh. 2014. Unal-nlp: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In *SemEval@COLING*.

Ryan Kiros, Yukun Zhu, Ruslan R. Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. In *SemEval@COLING*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*.

Yuhua Li, David McLean, Zuhair Bandar, James O'Shea, and Keeley A. Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18:1138–1150.

Nabin Maharjan, Rajendra Banjade, Dipesh Gautam, Lasang Jimba Tamang, and Vasile Rus. 2017. Dt team at semeval-2017 task 1: Semantic similarity using alignments, sentence-level embeddings and gaussian mixture model output. In *SemEval@ACL*.

Goutam Majumder, Partha Pakray, Alexander F. Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval@COLING*.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

George A. Miller. 1992. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI*.

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *ACL 2016*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Vasile Rus, Mihai C. Lintean, Rajendra Banjade, Nobal B. Niraula, and Dan Stefanescu. 2013. Semilar: The semantic similarity toolkit. In *ACL*.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *NIPS 2017*.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45:2673–2681.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *SemEval@ACL*.

Leslie N. Smith. 2015. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. Ecnu at semeval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *SemEval@ACL*.

Vedran Vukotic, Christian Raymond, and Guillaume Gravier. 2016. A step beyond local observations with a dialog aware bidirectional gru network for spoken language understanding. In *INTERSPEECH*.

Hao Wu, Heyan Huang, Ping Jian, Yuhang Guo, and Chao Su. 2017. Bit at semeval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In *SemEval@ACL*.

Jiang Zhao, Tiantian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *SemEval@COLING*.