# Rule-based vs. Neural Net Approach to Semantic Textual Similarity

**Linrui Zhang**
The University of Texas at Dallas
800 West Campbell Road; MS EC31,
Richardson, TX 75080 U.S.A
Linrui.zhang@utdallas.edu

**Dan Moldovan**
The University of Texas at Dallas
800 West Campbell Road; MS EC31,
Richardson, TX 75080 U.S.A
moldovan@hlt.utdallas.edu

## Abstract

This paper presents a neural net approach to determine Semantic Textual Similarity (STS) using attention-based bidirectional Long Short-Term Memory Networks (Bi-LSTM). To this date, most of the traditional STS systems were rule-based that built on top of excessive use of linguistic features and resources. In this paper, we present an end-to-end attention-based Bi-LSTM neural network system that solely takes word-level features, without expensive, feature engineering work or the usage of external resources. By comparing its performance with traditional rule-based systems against the SemEval 2012 benchmark, we make an assessment on the limitations and strengths of neural net systems as opposed to rule-based systems on STS.

## 1 Introduction

Semantic Textual Similarity (STS) is the task of determining the resemblance of the meanings between two sentences (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015; Agirre et al., 2016; Cer et al., 2017). For the sentence pairs below, on a scale from 0 to 5, (1) is very similar [5.0], (2) is somewhat similar [3.0] and (3) is not similar [0.2]:

1. Someone is removing the scales from the fish.

   A person is descaling a fish.

2. A woman is chopping an herb.

   A man is finely chopping a green substance.

3. The woman is smoking.

   The man is walking.

In STS tasks, the performance of traditional models relies highly on the usage of linguistic resources and hand-crafted features. For example, in SemEval 2012 Task 06: A Pilot on Semantic Textual Similarity (Agirre et al., 2012), the top three performers (Šarić et al., 2012; B*ä*r et al., 2012; Banea et al., 2012) all derived knowledge from WordNet, Wikipedia and other large corpora. In particular, Banea et al. built the models from 6 million Wikipedia articles and more than 9.5 million hyperlinks; B*ä*r et al. used Wiktionary, which contains over 3 million entries; and Šarić et al. used The New York Times Annotated Corpus that contains over 1.8 million news articles. Blanco and Moldovan (2013) proposed a model with semantic representation of sentences, which was considered to use the smallest external resources and features in 2015. However, their model still required WordNet with approximately 120,000 synsets and a semantic parser.

Complex neural network architectures are being increasingly used for learning to compute the semantic resemblances among natural language texts. To this date, there are two end-to-end neural network models proposed for STS tasks (Shao, 2017; Prijatelj et al., 2017), and both of them followed a standard sentence pair modeling neural network architecture that contains three components: a word embedding





*Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 12–17
Santa Fe, New Mexico, USA, August 20, 2018.

component that transforms words into word vectors, a sentence embedding component that takes word vectors as input and encodes the sentence into a single vector that represents the semantic meanings of the original sentence, and a comparator component that evaluates the similarity between sentence vectors and generates a similarity score.

In this paper, we modified and improved the modals proposed by Prijatelj et al. (2017) and Shao (2017), and proposed a Bi-LSTM neural network model as the representative of the neural net approach and evaluated it on the SemEval 2012 dataset. In the experimental section, we compared our system with the top three performers in SemEval 2012 using traditional rule-based models. Because neither Shao nor Prijatelj et al. considered attention mechanisms (Yang et al., 2016) in their systems, we specifically applied attention mechanisms to improve the performance of our system.

The goal of the paper is to illustrate that with well-designed neural network models, we can achieve competitive results (compared to traditional rule-based models) without expensive feature engineering work and external resources. We also make an assessment on the limitations of neural net systems as opposed to rule-based systems on STS.

## 2 Related Work

Determining textual similarity is relatively new as a stand-alone task since SemEval-2012, but it is often a component of NLP applications such as information retrieval, paraphrase recognition, grading answers to questions and many other tasks. In this section, we only list the works that are involved in our evaluation systems: the top performers in SemEval 2012 and recent neural network-based approaches in SemEval 2017.

The performance of the rule-based models (Šarić et al., 2012; B*ä*r et al., 2012; Banea et al., 2012) mostly rely on word pairings and knowledge derived from large corpora, e.g., Wikipedia. Regardless of details, each word in $sent_1$ is paired with the word in $sent_2$ that is most similar according to some similarity measure. Then, all similarities are added and normalized by the length of $sent_1$ to obtain the similarity score from $sent_1$ to $sent_2$. The process is repeated to obtain the similarity score from $sent_2$ to $sent_1$, and both scores are then averaged to determine the overall textual similarity. Several word to word similarity measures are often combined with other shallow features (e.g. n-gram overlap, syntactic dependencies) to obtain the final similarity score.

Shao (2017) proposed a simple convolutional neural network (CNN) models for STS. He used a CNN as the sentence embedding component to encode the original sentences into sentence-level vectors and generated a semantic difference vector by concatenating the element-wise absolute difference and the element-wise multiplication of the corresponding sentence vectors. He then passed the semantic difference vector into a fully connected neural network to perform regression to generate the similarity score on a continuous inclusive scale from 0 to 5. His model ranked 3rd on the primary track of SemEval 2017.

Prijatelj et al. (2017) wrote a survey on neural networks for semantic textual similarity. The framework of their model is similar to Shao's, but they explored various neural network architectures, from simple to complex, and reported the results of applying the combination of these neural network models within this framework.

## 3 System Description

Figure 1 provides an overview of our neural network-based model. The sentence pairs first pass through a pre-processing step described in subsection 3.1 to generate word embeddings. The attention-based Bi-LSTM models transform the word embeddings into sentence-level vectors described in subsection 3.2. In subsection 3.3, we use the same semantic difference vector as Shao to represent the semantic difference between the sentence-level vectors. Lastly, we pass the semantic difference vector into fully connected neural networks to generate the similarity score between the original sentence pairs.

### 3.1 Pre-processing

We first applied a simple NLP pipeline to the input sentences to tokenize them, remove punctuations and lower-case all the tokens. Second, we looked up the word embeddings from the pretrained 50-di-

mension GloVe vectors, and set non-existing words to zero vector. Third, we enhanced the word embeddings by adding a true/false (1/0) flag to them if the corresponding word appears in both sentences. Lastly, we unified the length of the inputs by padding the sentences.

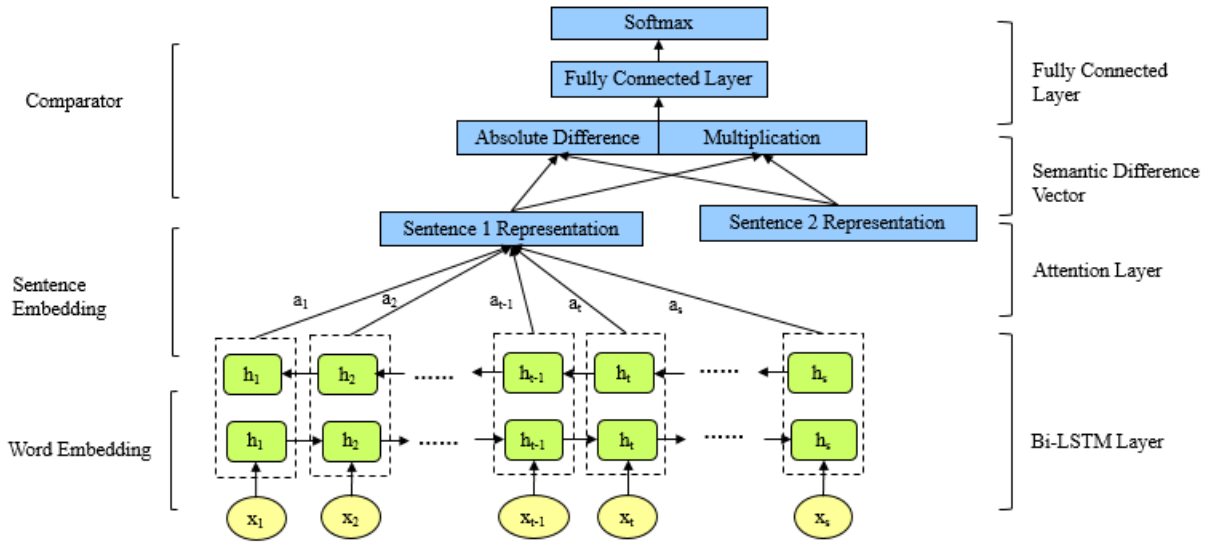


Figure 1: The structure of attention-based Bi-LSTM network for Semantic Textual Similarity.

### 3.2 Attention-based LSTM

Since sentences are sequences of words, and the order of the words matters, it is natural to use LSTMs (Hochreiter and Schmidhuber, 1997) to encode sentences into vectors. However, sometimes the backward sequence contains useful information as well, especially for long and unstructured sentences. Because of this, Irsoy and Cardie (2014) proposed Deep Bidirectional RNNs that can make predictions based on future words by having the RNN model read through the sentence backwards. In this section, we will first introduce a regular LSTM network and then extend it into a Bi-LSTM. At the end of this section, we will apply attention mechanisms to improve the performance of the system.

The traditional LSTM unit is defined by 5 components: an input gate, a forget gate, an output gate, a new memory generation cell and a final memory cell.

**The input gate** is to decide if the input $x_t$ is worth being preserved based on the input word $x_t$ and the past hidden state $h_{t-1}$.

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \tag{1}$$

**The forget gate** $f_t$ makes an assessment on whether the past memory cell is useful to compute the current memory cell.

$$f_t = \sigma(w^{(f)}x_t + U^{(f)}h_{t-1}) \tag{2}$$

**The output gate** is to separate the final memory $c_t$ from the hidden state $h_t$.

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \tag{3}$$

**The new memory generation cell** is used to generate a new memory $\tilde{c}_t$ by input work $x_t$ and the past hidden state $h_{t-1}$.

$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \tag{4}$$

**The final memory cell** produces the final memory $c_t$ by summing the advice of the forget gate $f_t$ and input gate $i_t$.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{5}$$

$$h_t = o_t \cdot tanh(c_t) \quad (6)$$

A Bi-LSTM could be viewed as a network that maintains two hidden LSTM layers together, at each time-step *t*, one for the forward propagation and another for the backward propagation. The final classification results are generated through the combination of the score results produced by both hidden layers. The mathematical representation of a simplified Bi-LSTM is shown as follows:

$$\vec{h}_t = f(\vec{W}x_t + \vec{V}\vec{h}_{t-1} + \vec{b}) \quad (7)$$

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}\overleftarrow{h}_{t-1} + \overleftarrow{b}) \quad (8)$$

$$\hat{y}_t = g(Uh_t + c) = g(U[\vec{h}_{t-1}; \overleftarrow{h}_{t-1}] + c) \quad (9)$$

where $\hat{y}_t$ is the final predication. The symbols → and ← are indicating directions. The rest of the terms are defined the same as in regular LSTM neural networks. *W, U* are weight matrices that are associated with input $x_t$ and hidden states $h_t$. *U* is used to combine the two hidden LSTM layers together, *b* and *c* are bias term. *g(x)* and *f(x)* are activation functions.

Not all words contribute equally to the representation of the sentence meaning; thus, we extract words that are more informative to the sentence and aggregate these words to the sentence-level vector by applying the attention mechanism. Specifically:

First, we feed the hidden state $h_t$ through a one-layer MLP to get $u_t$, and $u_t$ could be viewed as a hidden representation of $h_t$.

$$u_t = tanh(Wh_t + b) \quad (10)$$

Second, we multiply $u_t$ with a context vector $u_w$, and normalized the results through a softmax function to get the importance weight $a_t$ of each hidden state $h_t$. The context vector could be seen as a high-level vector to select informative word in the sentence (Sukhbaatar et al., 2015) and it will be jointly learned during the training process.

$$a_t = \frac{exp(u_t^T u_w)}{\sum_t exp(u_t^T u_w)} \quad (11)$$

Lastly, the final state *S* is a sum over of the hidden states and its the importance weights.

$$S = \sum_t a_t h_t \quad (12)$$

### 3.3 Semantic Difference of Sentences

We used the same semantic difference vector as Shao, by concatenating the element-wise absolute difference and the element-wise multiplication of the corresponding sentence-level embedding pairs.

The generated semantic difference vector is passed through a two-layer fully connected neural networks with a softmax function as the output layer and generated a probabilistic distribution over the six similarity labels used in the SemEval 2012 task. We multiplied it with a constant matrix with integers from 0 to 5 to transfer the probabilistic distribution into a float number as the final semantic similarity score between the original sentence pairs.

## 4 Experiment

### 4.1 Corpus

We evaluated our system on the corpora used in SemEval 2012 Task 06: A Pilot on Semantic Textual Similarity. It contains five corpora: (1) MSRvid, short sentences for video descriptions; (2) MSRpar, long sentences of paraphrases; (3) SMTeuroparl, output of machine translation systems and reference translations; (4) OnWN, OntoNotes and WordNet glosses; and (5) SMTnews, output of machine translation systems in the news domain and gold translations. In corpus (1) to (3), both training and testing data are provided, and corpus (4) and (5) are surprise data (new domain data) and only testing data are provided. For more details about the corpus, please refer to Agirre et al. (2012).

We followed the training and testing splits of the original corpus. Since corpus (4) and (5) do not have corresponding training data, we used the training data of corpus (2) as the training data for corpus (4) since both corpus contains long and hard to parse sentences, and we used the training data of corpus (3) as the training data for corpus (5), since both corpus contains ungrammatical sentences.

### 4.2 Experiment Results

We used the Pearson correlation coefficient to evaluate the performance. We introduced two neural network models: a regular LSTM model and a Bi-LSTM model and, for each model, we also demonstrated their performance with attention mechanisms. Table 1 shows the results of the neural network-based systems and the traditional rule-based systems on in-domain data (corpus 1 to 3) and out-of-domain data (corpus 4 and 5).

| **System** | | **MSRvid** | **MSRpar** | **SMTeuroparl** | **OnWN** | **SMTnews** |
|---|---|---|---|---|---|---|
| LSTM | Basic | 0.7774 | 0.5278 | 0.2787 | 0.4519 | 0.2071 |
| | +attention | 0.7851 | 0.5891 | 0.3492 | 0.4773 | 0.2635 |
| Bi-LSTM | Basic | 0.7661 | 0.5258 | 0.3993 | 0.4591 | 0.3298 |
| | +attention | 0.7762 | 0.6210 | 0.4368 | 0.5607 | 0.3976 |
| Bär et al., 2012 | | 0.8739 | 0.6830 | 0.5280 | 0.6641 | 0.4937 |
| Šarić et al., 2012 | | 0.8620 | 0.6985 | 0.3612 | 0.7049 | 0.4683 |
| Banea et al., 2012 | | 0.8750 | 0.5353 | 0.4203 | 0.6715 | 0.4033 |

Table 1: The Person correlation coefficient of our system and the top three performers in SemEval 2012 benchmark.

### 4.3 Results Analysis

From the results, we could observe that: (1) The overall performance of the rule-based model is still slightly better than the neural network-based approach. However, we must note that the neural network models are end-to-end models that do not use complicated linguistic features and resources. (2) The neural network-based approaches are better at handling long sentences, whereas the rule-based systems are good at handling short sentences. The reason is that the performance of the traditional rule-based models greatly relies on the extraction of features, however, long sentences are usually hard to parse. The errors that occur in the feature extraction step will propagate until the end and decrease the performance of the system. Whereas the neural network models only take word-level features and do end-to-end training, so they do not have this "error propagation" issue. Besides, since we add attention mechanisms, the system could aggregate the influence of the informative words and ignore the unimportant words in long sentences. From the results we observe that our system beats the third-ranked performer on the MSRpar corpus, and the second- and third-ranked performers on the SMTeuroparl corpus, which contains mainly long sentences. (3) The regular LSTM model performs poorly on the SMTeuroparl corpus, but the Bi-LSTM dramatically increases the performance (ranking just after the top performer with rule-based systems). The reason is that in the SMTeuroparl corpus, one sentence in the sentence pair is usually ungrammatical. Regular LSTM can only capture the forward sequential information of sentences, so it will miss some information if the sentences are unstructured. However, the Bi-LSTM model can compensate for this missing information by capturing the backward sequential information as well, and this makes the system more robust when handling ungrammatical sentences. (4) The traditional rule-based models show a huge advantage over neural network-based models on new domain datasets. The reason is that the neural-network models are supervised models that mostly depend on the training data, and when transferred to new domains lacking training data, the performance of the system drops dramatically. On the other hand, rule-based systems rely mostly on word pairings and linguistic resources that are not as dependent on training data.

## References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In Proceedings of the First Joint Conference on Lexical and Computational SemanticsVolume

1*: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

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*, volume 1, pages 32–43.

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)*, pages 81–91.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 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)*, pages 252–263.

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)*, pages 497–511.

Carmen Banea, Samer Hassan, Michael Mohler, and Rada Mihalcea. 2012. Unt: A supervised synergistic approach to semantic text similarity. In *Proceedings of 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*, pages 635–642. Association for Computational Linguistics.

Daniel B*ä*r, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of 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*, pages 435–440. Association for Computational Linguistics.

Eduardo Blanco and Dan Moldovan. 2013. A semantically enhanced approach to determine textual similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1235–1245.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint* arXiv:1708.00055.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 720–728.

Derek S Prijatelj, Jonathan Ventura, and Jugal Kalita. 2017. Neural networks for semantic textual similarity. In *Proceedings of the 14$^{th}$ International Conference on Natural Language Processing*, pages 456-465.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of 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*, pages 441–448. Association for Computational Linguistics.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, Montréal, Canada, pages 2440–2448.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pages 1480–1489.