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-
cently used successfully in sentence similarity (Ne
culoiu et al., 2016). Siamese architectures are good in
these tasks because, when the inputs are of the same kind, it makes sense to
to a similar model to process similar inputs. In
this way the networks will have representation vectors with the same semantics, mak-
ing 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 sev-
earal architectures which use Siamese neural
networks for STS. The rest of the paper is
organized as follows. Section 2 briefly de-
scribes several approaches used to measure
sentence similarity focusing more on Siamese
neural networks. Section 3 contains informa-
tion about the settings of the experiments car-
rried 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 pro-
posed a large number of such metrics. Before
the shift of interest in neural networks, most
of the proposed methods relied heavily on fea-
ture engineering. A typical example is (Gu-
pta et al., 2014a) which employed 20 linguis-
tic features fed into a support vector machine re-
gressor. 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 sev-
eral learning algorithms like support vec-
tor machine regressor, Random Forest, Gradi-
ent boosting etc (Zhao et al., 2014). With the
introduction of word embedding models, re-
searchers focused more on neural representa-
tion for this task. Many of the leading teams
in the STS task at Semeval 2017 used some
kind of neural network architecture which em-
ployed word embeddings (Shao, 2017). As an
example, Maharjan et al. (2017) used an en-
semble of traditional machine learning models
and deep learning models in their top perform-
ing system at Semeval 2017 STS task.

There are two main approaches which em-
ploy neural representation models: supervised
and unsupervised. Unsupervised approaches
use pretrained word/sentence embeddings di-
rectly for the similarity task without train-
ing 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 dimen-
sional 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 net-
work (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 Recur-
rent Neural Network (RNN) encoder-decoder
with Gated Recurrent Unit (GRU) activa-
tions. They attempt to reconstruct the im-
mediately 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
(2015) which generalize the order-sensitive chain-structure of stan-
dard LSTMs to tree-structured network topo-
lologies. 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 syn-
tactic 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 dataset 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/
<table>
<thead>
<tr>
<th>Approach</th>
<th>$\tau$</th>
<th>$\rho$</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois-LH</td>
<td>0.7993</td>
<td>0.7538</td>
<td>0.3692</td>
</tr>
<tr>
<td>(Lai and Hockenmaier, 2014)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNAL-NLP</td>
<td>0.8070</td>
<td>0.7489</td>
<td>0.3550</td>
</tr>
<tr>
<td>(Jiménez et al., 2014)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meaning Factory</td>
<td>0.8268</td>
<td>0.7721</td>
<td>0.3224</td>
</tr>
<tr>
<td>(Bjerva et al., 2014)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECNU</td>
<td>0.8414</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>(Zhao et al., 2014)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skip-thought+COCO</td>
<td>0.8655</td>
<td>0.7995</td>
<td>0.2561</td>
</tr>
<tr>
<td>(Kiros et al., 2015)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dependency Tree-LSTM</td>
<td>0.8676</td>
<td>0.8083</td>
<td>0.2532</td>
</tr>
<tr>
<td>(Tai et al., 2015)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConvNet</td>
<td>0.8686</td>
<td>0.8047</td>
<td>0.2606</td>
</tr>
<tr>
<td>(He et al., 2015)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bi-directional LSTM†</td>
<td>0.8743</td>
<td>0.8251</td>
<td>0.2391</td>
</tr>
<tr>
<td>GRU + Capsule + Flatten†</td>
<td>0.8786</td>
<td>0.8286</td>
<td>0.2301</td>
</tr>
<tr>
<td>MALSTM (Baseline)</td>
<td>0.8822</td>
<td>0.8345</td>
<td>0.2286</td>
</tr>
<tr>
<td>(Mueller and Thyagarajan, 2016)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM: Adagrad†</td>
<td>0.8831</td>
<td>0.8364</td>
<td>0.2195</td>
</tr>
<tr>
<td>GRU + Attention†</td>
<td>0.8843</td>
<td>0.8372</td>
<td>0.2163</td>
</tr>
<tr>
<td>LSTM + Attention†</td>
<td>0.8886</td>
<td>0.8386</td>
<td>0.2142</td>
</tr>
<tr>
<td>Bi-directional GRU†</td>
<td>0.8896</td>
<td>0.8390</td>
<td>0.2125</td>
</tr>
<tr>
<td>GRU†</td>
<td>0.8901</td>
<td>0.8396</td>
<td>0.2112</td>
</tr>
</tbody>
</table>

Table 1: Pearson correlation ($\tau$), Spearman correlation ($\rho$), and Mean Square Error (MSE) for the SICK test set.

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-

![Basic structure of the Siamese neural network. Unit A is changed over the architectures.](image-url)
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.

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,

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

Table 2: Pearson correlation (τ) for STS2017 test set.
<table>
<thead>
<tr>
<th>Sentence 1</th>
<th>Sentence 2</th>
<th>GOLD</th>
<th>LSTM</th>
<th>GRU</th>
</tr>
</thead>
<tbody>
<tr>
<td>The people are walking on the road beside a beautiful waterfall</td>
<td>The people are walking on the road beside a waterfall, which is beautiful</td>
<td>0.9750</td>
<td>0.5260</td>
<td>0.9569</td>
</tr>
<tr>
<td>The woman is frying a chop of breaded pork</td>
<td>The woman is frying a breaded pork chop</td>
<td>0.9250</td>
<td>0.5278</td>
<td>0.8561</td>
</tr>
<tr>
<td>A white dog is standing on the leaves on the ground</td>
<td>A dog, which is white, is standing on fallen leaves</td>
<td>0.9500</td>
<td>0.3611</td>
<td>0.7618</td>
</tr>
<tr>
<td>The man is erasing the other man’s work from the board</td>
<td>The man is erasing the drawing on the board</td>
<td>0.7500</td>
<td>0.5128</td>
<td>0.7584</td>
</tr>
</tbody>
</table>

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

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

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 Janice Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In SemEval@COLING.


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

Luísa 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.


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


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*.


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*.


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*.


Vedran Vukotic, Christian Raymond, and Guillemme 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.