Siamese Network with Soft Attention for Semantic Text Understanding

Adebayo Kolawole John
Department of Computer Science,
University of Torino
Torino 10149, Piemonte, Italy
kolawolejohn.adebayo@unibo.it

Luigi Di Caro
Department of Computer Science,
University of Torino
Torino 10149, Piemonte, Italy
dicaro@di.unito.it

Guido Boella
Department of Computer Science,
University of Torino
Torino 10149, Piemonte, Italy
guido@di.unito.it

ABSTRACT

We propose a task independent neural networks model, based on a Siamese-twin architecture. Our model specifically benefits from two forms of attention scheme, which we use to extract high level feature representation of the underlying texts, both at word level (intra-attention) as well as sentence level (inter-attention). The inter attention scheme uses one of the text to create a contextual interlock with the other text, thus paying attention to mutually important parts. We evaluate our system on three tasks, i.e. Textual Entailment, Paraphrase Detection and Answer-Sentence selection. We set a near state-of-the-art result on the textual entailment task with the SNLI corpus while obtaining strong performance across the other tasks that we evaluate our model on.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Computer systems organization** → **Siamese Neural Networks**; *Natural Language Understanding*; Deep Learning; Textual Entailment; Paraphrase Detection; Answer Sentence Selection;

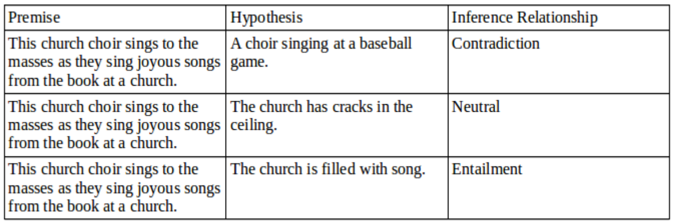
KEYWORDS

Neural Networks, Siamese Architecture, Textual Entailment, Paraphrase Detection, Answer Sentence Selection, Information Retrieval

ACM Reference format:
Adebayo Kolawole John, Luigi Di Caro, and Guido Boella. 2017. Siamese Network with Soft Attention for Semantic Text Understanding. In *Proceedings of Semantics2017, Amsterdam, Netherlands, September 11–14, 2017,* 8 pages.
https://doi.org/10.1145/3132218.3132236

1 INTRODUCTION

How do we ascertain the level of similarity of two text snippets? If indeed, two text snippets deemed to be semantically related, how do we identify the type of logical relationship that they have? Measuring text similarity is a fundamental problem in language modeling and plausible solutions to the questions above have been proposed in previous work which addresses tasks like semantic textual similarity or semantic relatedness (STS), paraphrase detection (PD), textual entailment (RTE) and question answering (QA). Specifically, STS, PD, and RTE are usually grouped under a suite of sentence modeling task called natural language inference (NLI).



Premise	Hypothesis	Inference Relationship
This church choir sings to the masses as they sing joyous songs from the book at a church.	A choir singing at a baseball game.	Contradiction
This church choir sings to the masses as they sing joyous songs from the book at a church.	The church has cracks in the ceiling.	Neutral
This church choir sings to the masses as they sing joyous songs from the book at a church.	The church is filled with song.	Entailment

Figure 1: Sample texts from SNLI Corpus

In RTE, given a premise and a hypothesis, the goal is to identify if the meaning of the hypothesis can be inferred from the premise [5, 6]. If this is true, we say that the premise entails the hypothesis, otherwise, there is a contradiction or the relationship is neutral. Some snippets from SNLI corpus are shown in figure 1, with an example each of contradiction, neutral and entailment inference relationship. In STS, we quantify how semantically related two given texts are, usually, the notion of similarity is graded [2, 13], i.e. on a low-to-high scale of 1-5. PD also involves examining two sentences and determining whether they have the same meaning or not [4, 24], while for QA, the goal is to match a question to some answer containing sentence(s) [10, 28]. Even though these tasks appear simple on the surface, they are quite challenging due to the variability of linguistic expressions and therefore, solutions often require stringent feature engineering. Also, conventional machine learning techniques often rely on the use of a lot of external language resources, e.g. semantic nets, etc., in order to come up with models that scale up in terms of performance.

With recent exploits of neural network models in natural language modeling tasks like machine translation [3, 27] and QA [34, 35] and text classification [39, 40], researchers have used a kind of recurrent networks such as long short-term memory (LSTM) [12] and recurrent neural networks (RNN) [17, 25] to encode text pair independently as embedding vectors which are then combined and fed to a multilayer perceptron (MLP) network for making semantic relationship decision between the text pair.

Bowman et. al., [6] introduced the *SNLI* corpus and used LSTM to encode both the premise and hypothesis while obtaining a modest 77.6% accuracy. This was improved by [22] who introduced some word-by-word attention to capture the interaction between

Semantics2017, September 11–14, 2017, Amsterdam, Netherlands
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5296-3/17/09…$15.00
https://doi.org/10.1145/3132218.3132236

words as well as the important words to focus on. Most importantly, *Attention* schemes have proven to enhance the performance of neural networks. Bahdanau et. al.,[3] demonstrate this with machine translation while the work of [15, 16, 19, 29] also employ attention across varying tasks.

Wang et. al., [32] proposed a multi-perspective matching of sentences and evaluated their model on the task of RTE, PD and Answer Sentence selection. A unifying identity among the majority of the existing systems is that other than the incorporation of attention techniques, the encoded texts share fewer parameters and interaction for they are based on sentence encoding paradigm [32].

Furthermore, even the attention schemes which relies on the sentential representation[1] downplays the syntactic relationship and interplay between the words in each sentence and may not explicitly capture the contextual information in those sentences. However, the inter-sentence relationship and contextual properties are important because the meaning of a word is context dependent.

In this work, we introduce a more fine-grained attention, both in word and sentence level. Our contribution is how we employed attention to generate features for each text, using a Siamese architecture. Our goal is to develop a baseline model that generalizes well across different tasks, while also modeling both 'intra' and 'inter' contextual relationship between words. By inter and intra attention, we mean the inter-sentence and intra-sentence attention. Where intra-sentence attention measures the importance of each word within a sentence based on its similarity with a word-level context vector and it is then able to select the most important words. This is particularly important since different words in a text are differentially informative while also being context dependent[37]. On the other hand, inter-sentence attention aims to create an interaction between two sentences, by measuring the importance of the words in one sentence, conditioned on the words in the other sentence, i.e., using a sentence-level context vector. Our attention scheme allows us to extract more salient features which give a better representation of the texts being compared. This is then used for onward classification in all the tasks studied in this work. As we will see later, we evaluate our model on three different tasks. Apart from minor alteration needed for proper prediction, we did not essentially fine-tune our model for any specific task. In spite of this versatility, we obtained good results which are at par with or exceeding some state of the art systems. The tasks studied in this work include recognizing textual entailment, paraphrase detection and answer sentence selection.

In the remaining parts of the paper, we elucidate some related works, followed by the description of our model. Next, we describe the experiments and give a detailed analysis of the results obtained per task.

2 RELATED WORK

A large body of work based on neural networks applied to text pair have since built up with the availability of bigger datasets, e.g. SNLI [6]. These systems are mostly based on sentence encoding [6, 7, 22], Where LSTMs are often used to embed premise and hypothesis in the same vector space. This approach enhances parameter sharing, which is then propagated to some neural network components. Other techniques like the Attention mechanism [15, 16, 19], Extended Memory Structure [8, 18] and Factorization-based Matching [32] builds on the former by providing more interaction between embedded sentences.

While these systems have reported impressive results, they often exhibit a deep structure of sentence modeling while ignoring contextual dependencies between the words of each sentence. Furthermore, they usually have excessive trainable parameters.

Bowman et. al., [6] introduced the SNLI corpus, which contains 570k human annotated text pairs. They employed a lexical classifier as the baseline for their LSTM encoding-based network. The work in [22, 38] employs attention techniques with LSTM and convolution neural networks (CNN) in order to improve the performance of their models. In the work of Rocktaschel et. al., [22], the two texts were read together, an LSTM first reason over the sequence of tokens in the text while the second LSTM is reasoning on the hypothesis sequence. The second LSTM is conditioned by the first one since its memory is initialized by the output (i.e., the last cell state) of the last hidden state of the first LSTM when reading each input from the hypothesis. A Bidirectional LSTM was also used in this manner to create dual-attention.

Parikh et. al., [19] improved on the attention mechanism of [22] by introducing two components, i.e., *compare* and *aggregate*. The former compares aligned phrases in the premise with that of the hypothesis and vice versa, using a Feedforward Neural Network. The resulting vectors are further summed over in the Feedforward Neural Network component. Cheng et. al., [8] utilized a type of LSTM with an enhanced memory, called the LSTMN. Similar to the Memory Networks of [35]. They used two attention schemes, i.e., a *shallow fusion* which considers only the intra-attention and a *deep fusion* which consider both attention between the hypothesis and text. The *deep fusion* achieved superior performance. Overall, both models achieved near state-of-the-art results on the tasks of textual entailment, sentiment analysis and language modeling without the need to specially fine-tune the model for each of these tasks. A question answering model which use LSTM to project both the question and answer into the same semantic space was proposed in [28].

Wang et. al., [32] introduced a Bilateral Multi-Perspective Matching model which shares some similarity with our model. Their model, which is the state of the arts on SNLI shares many commonalities with previous works in that they also used bidirectional-LSTM. The innovative part of their work is the matching scheme. First, a BiLSTM each encodes the premise and hypothesis, then, using any of the four matching function i.e., Full matching, Maxpooling, Attentive and Max-attentive matching, each time-step of the premise is matched against every time-step of hypothesis and vice-versa. Another BiLSTM then combines the result before passing through a Fully Connected layer (FC) for classification.

Our proposed approach is Siamese in nature since we have two parallel models, each processing a text and its corresponding hypothesis and vice versa, i.e., the first model takes both the text and the hypothesis and then outputs a feature vector of the text conditioned on the hypothesis. Likewise, the second model also takes in the text and the hypothesis and outputs a feature vector of

[1] Here, the hidden state from the last time step of a sentence is used to attend to each time step of the other sentence, and such attention tends to be bias towards the latter words of the first sentence.

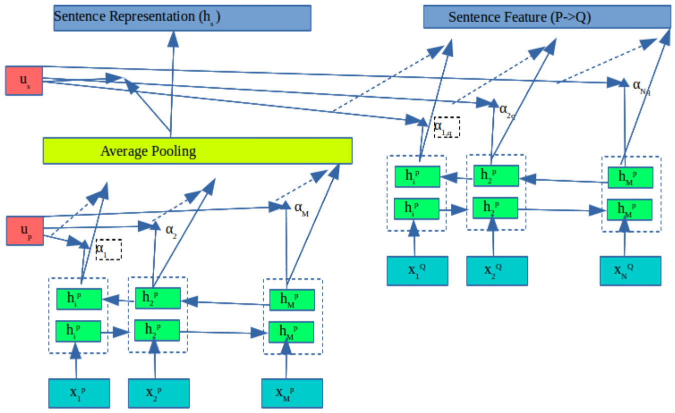


Figure 2: High-level architecture of the proposed model, showing interation P–>Q

the hypothesis conditioned on the text. For the sake of clarity, we denote the first model as the *text model* and the second model, the *hypothesis model*.

Each model utilizes LSTM to obtain a sentence level representation of the text. However, we introduce a word level intra-attention scheme that captures the contextual dependencies between the words in a sentence. Also, instead of using the last hidden state as the sentence representation, we use an average of all the hidden states at each time-steps. The inter-attention component uses the sentence representation obtained from the first text to attend to every time-steps of the second text[2] and vice versa, thus creating an interaction within the two text piece. The models also share parameters during training. Then, we obtain two feature vectors, each from the *text model* and the *hypothesis model* which summarizes their input text. These feature weights are further projected to a FC layer for classification. We give a detailed description of our model in the next section.

3 METHODS

Our approach can be divided into three parts, i.e., word encoding, attention scheme and lastly, sentence encoding and features generation. A high-level representation of the *text model*, which is just a part of the Siamese network is shown in figure 2. Note that due to space, the diagram only shows a child-model of the Siamese model, i.e. capturing the high-level representation of the *text* conditioned on the *hypothesis*.

For all the tasks, we assume two texts P = (p_1,....,p_i,....,p_M) and Q = (q_1,....,q_j,....,p_N) both of length M and N respectively. Also, a label y ∈ Y is given which shows the relationship depending on the task. Usually, y is a binary output for PD task and QA task. P is typically the premise in an RTE task, the question in an Answer-Sentence selection task and the passage in a QA task etc. The same thing applies for Q. In anyway, the data representation is task specific, but they follow the same format of (P,Q,y) triples. The goal then, is to estimate the conditional probability Pr(y | P, Q) based on the training set, and predicting the relationship for testing samples by $\mathrm{y}^* = \arg\max_{y \in Y} \Pr(\mathrm{y} \mid \mathrm{P}, \mathrm{Q})$.

We need to state that our model has a Siamese structure [14] which we represent as two parallel models, i.e., *text model* and *hypothesis model*. Each of the model takes in both P and Q at the same time. For the sake of brevity and as shown in figure 2, we describe the structure of the system with respect to just one child-model of the Siamese, e.g., the *text model*, it should be noted that this structure or computation applies to both models in the Siamese network. Generally, we used a Bi-directional LSTM to encode the text and obtain a representation. The reader is referred to [12] for details about LSTM but the state transitions is given in equation 1. Given (x_t as the input vector at time step t, σ represents sigmoid activation function and ⊙ the element-wise multiplication. LSTM has *input gate* i_t, a *forget gate* f_t, an *output gate* o_t, a *memory cell* c_t and a *hidden state* h_t. The u_t is a tanh layer which creates a vector of new candidate values that could be added to the state.

$$\begin{aligned}
i_t &= \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right),\\
f_t &= \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right),\\
o_t &= \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right),\\
u_t &= \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot \tanh c_t \qquad (1)
\end{aligned}$$

We now proceed to describe the components of the proposed system.

3.1 Word Encoding

Here, we represent each word in the sentences P and Q with a d-dimensional vector, where the vectors are obtained from a word embedding matrix. Generally, we use the *GLOVE* 300-dimensional vectors obtained with 840 billion words trained [20]. A Bi-directional LSTM is then used in order to obtain contextual information between the words. A Bi-directional LSTM is essentially composed of two LSTMs, one capturing information in one direction from the first time step to the last time-step while the other captures information from the last time-step to the first. The outputs of the two LSTMs are then combined to obtain a final representation which summarizes the information of the whole sentence. Equations (2) and (3) describes this computation.

$$\begin{aligned}
\overrightarrow{h_i^p} &= \overrightarrow{LSTM}(\overrightarrow{h_{i-1}^p}, P_i), \quad i \in [1, \ldots, M]\\
\overleftarrow{h_i^p} &= \overleftarrow{LSTM}(\overleftarrow{h_{i-1}^p}, P_i), \quad i \in [M, \ldots, 1] \qquad (2)
\end{aligned}$$

$$h_i^f = [\overrightarrow{h_i^p}; \overleftarrow{h_i^p}] \qquad (3)$$

[2] Here, we abuse the alternating usage of the words 'text' and 'hypothesis' as generally used in the Textual Entailment task to mean the same thing with the words 'first text' and 'second text' respectively, as may be applicable to answer-sentence selection or paraphrase detection tasks.

3.2 Attention Scheme

Attention is a way of focusing specially on some important parts of an input, and has been used extensively in some language modeling tasks [3, 19, 29]. Essentially, it is able to identify the parts of a text that are most important to the overall meaning of the text. We use two forms of attention, i.e., the *inter* and *intra* attention. Our intra attention focuses on important words within the same text. Specifically, such important words can now be aggregated to compose the meaning of that text. On the other hand, the inter attention tries to attend to important words in the second text conditioned on the intra-attention representation of the first text. Similar to [3], we use intra-attention to obtain the sentence representation as shown in equation (6) by obtaining the average of the sum of all attention weighted hidden states. Initially, the encoded sentence (see equation (3)) is first passed through a single layer MLP to get a hidden representation u$_i$ which is then weighted with the attention vector α_i across the time-steps. The weights from α_i sum up to 1, and are used to compute a weighted average of the last hidden layers generated after processing each of the input words. Note that we take the sentence representation as the average of the summed weighted hidden states (see equation (6)).

$$u_i = \tanh(W_p h_i^f + b_p) \tag{4}$$

$$\alpha_i = \frac{\exp(u_i^M u_t)}{\sum_i \exp(u_i^M u_t)} \tag{5}$$

$$h_s = \sum_i \alpha_i h_i^f \tag{6}$$

The inter attention follows a similar pattern and we actually use it to extract a feature vector for the sentence using equations (7) - (9). Specifically, what this means is that each model from the Siamese network takes in sentences P and Q. The model for P, uses intra-attention to aggregate its important words, then use this sentence representation to create another attention (inter) which is conditioned on each time steps of Q. Thus, we are able to model an interaction (P ⟶ Q) and extract important features for text P obtained with equation (9). The model for text Q also takes in both Q and P and performs the same computation to obtain the feature vector for text Q, also obtained with equation (9), based on its interaction (Q ⟶ P).

$$u_s = \tanh(W_s h_s + b_s) \tag{7}$$

$$\alpha_s = \frac{\exp(u_s^N u_t)}{\sum_q \exp(u_q^N u_t)} \tag{8}$$

$$s_p = \sum_q \alpha_s h_s \tag{9}$$

Recall that we only show the computation for the *text model* with interaction P ⟶ Q. Here, q ∈ [1,.....,N] signifies the time-steps of Q, where the hidden sentence representation h$_s$ of P is used to attend to each time-step q. Analogously, the same thing applies when the computation for *hypothesis model* is done.

The ensuing representations S$_p$ and S$_q$ can be regarded as a high level representation of both texts P and Q which are then concatenated and propagated to a MLP classifier as shown in equation (10) depending on the task. The predicted class is obtained from the probability distribution given in equation (10) . For training, we use multi-class cross-entropy loss with dropout regularization [26]

$$\hat{y} = H([s_p; s_q]) \tag{10}$$

The choice of classifier depend on the task

$$\hat{y} = \ arg\ Max_i \hat{y}_i \tag{11}$$

We trained our model with the cross-entropy loss given in equation 12 where θ_F, θ_G , θ_H are parameters to be learned.

$$L(\theta_F, \theta_G, \theta_H) = \frac{1}{J} \sum_{j=1}^{J} \sum_{c=1}^{C} y_c^{(j)} \log \frac{\exp(\hat{y}_c)}{\sum_{c=1}^{C} \exp(\hat{y}_c)} \tag{12}$$

4 EXPERIMENTS

We evaluate our model on three different tasks, namely textual entailment, paraphrase detection and question answering. Each of these tasks uses different datasets. Moreover, we show a bias towards the recognizing textual entailment (RTE) task in our discussion, since the datasets used are relatively huge. Generally, we used the 300-dimensional GloVe word vectors pre-trained from the 840B Common Crawl corpus to initialize the word embeddings [20]. Throughout the network, the weights of the word embedding remain fixed in order to reduce the number of trainable parameters. Out-of-vocabulary (OOV) words were randomly assigned some embedding vectors. We use two Bi-directional LSTMS, one for each model of the Siamese network. Hidden size of our bi-directional LSTM was set as 300. We use ADAM, a stochastic optimizer with learning rate set at 0.01 and a decay value of 1e-4 as well as momentum of 0.9. Wherever we applied Dropout, we ensure a fixed size of 0.2. Early stopping was utilized to avoid over-fitting. The Patience value for monitoring when the loss of the development test set stops decreasing was set to 4. For all the experiments, we pick the model with the best performance on the development set, and then evaluate it on the test set. Where there is no standard train-dev-test split (e.g., Quora and MNLI datasets), we split the original data into train, dev and test set according to this ratio respectively: 80:10:10. Overall, our model has 3.2m trainable parameters excluding the word embeddings.

4.1 Datasets

For the task of textual entailment (RTE), we evaluate our system on the more popular SNLI [6] corpus as well as the newly introduced MultiNLI[3] corpus which is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. The corpus is modeled on the SNLI corpus, but differs in that it covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation. Unlike the SNLI, the authors of the corpus do not release a standardize test split, rather, two dev-sets (matched and mismatched) were provided. We interchange these two dev-sets such that when one is used as dev, the other is used as test set respectively and vice versa. For the task of Paraphrase detection, we report our evaluation on Quora dataset [4]. Quora contains 404,351 valid text pairs and we maintain our train-test-split ratio already highlighted before. The second dataset for paraphrase detection is the Microsoft Research Paraphrase Corpus

[3] http://www.nyu.edu/projects/bowman/multinli/

[4] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

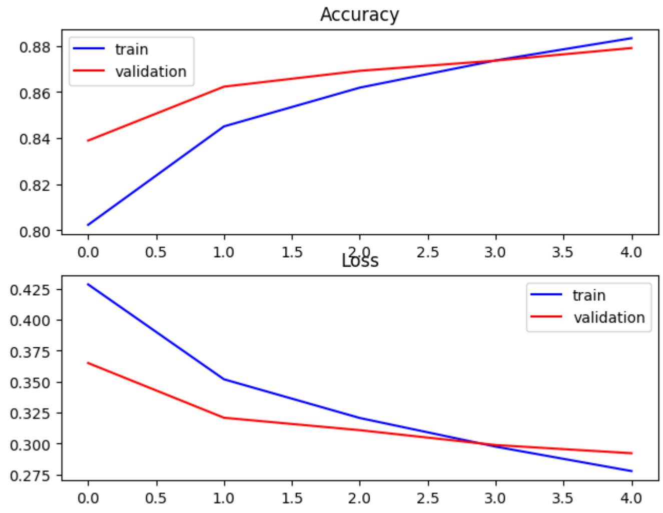


Figure 3: Plots of Accuracy and Loss of Train and Dev set of SNLI

which also contains 3577 train text pairs, 501 validation set and 1726 test set. The task on answer sentence selection was evaluated with the WikiQACorpus.

Models	Accuracy
Bowman et. al.,[6]	77.6
Rocktaschel et. al.,[22]	83.5
Liu et. al.,[16]	85.0
Shouohang et. al.,[30]	86.1
Cheng et. al.,[8]	86.3
Parikh et. al.,[19]	86.8
Sha et. al., [23]	87.5
Chen et. al.,[7]	87.7
Wang et. al.,[32]	**88.8**
This paper	88.2

Table 1: RTE result on SNLI dataset

Test Split	Accuracy
Train	80.0
Dev(matched)	72.0
Test (Mismatched)	**74.0**
Dev(Mismatched)	72.4
Test (Matched)	**74.2**

Table 2: RTE result on MultiNLI dataset

Premise	Hypothesis	Gold Label	Prediction
She's a , she works in a warehouse now, actually	She is at the warehouse	Contradiction	Entailment
As with many new technologies that have developed since 1970's, adoption of innovative equipment is still occuring in fits and starts and depends on a given firm's size and mix of products	A firm's size and product variety depend on whether or not they adopt innovative equipment	entailment	contradiction
From home work to modern manufacturing	Modern manufacturing has changed over time	entailment	neutral

Figure 4: Observations from MultiNLI dataset

4.2 Experiment on Textual Entailment

Here, we evaluate our system on two different datasets, i.e., the SNLI corpus as well as the MultiNLI dataset, a multi-genre corpus which was recently introduced. The results obtained from both datasets are provided in tables 1 and 2. We can see that our system achieved an impressive accuracy of 88.2 which is very close to best performing system. As a matter of fact, our system has fewer parameter compared to the system of [32] even though we did not report this analysis in the table. Most of the best performing systems employ one form of attention or the other. A plot of the accuracy and loss values for both the train and development set is shown in figure 3. We can see a steep growth curve for the accuracy plot which implies that our the model does not overtly over-fits. Parameters estimation and results of competing models are also available on SNLI leadeboard website[5]

For MultiNLI corpus, as shown in table 2, we do not compare our system with any known system simply because as at the time of writing this article, there is currently no published work which has been evaluated on this data. However, the curator of the dataset has introduced a public Leaderboard for reporting evaluation result on this data. Moreover, the current best result on the leaderboard [6] for this dataset is 74.4 % accuracy, which is just a slight improvement on our performance, i.e., 74.2 % accuracy [7]. Overall, our system did not scale optimally on this dataset. We assume that this dataset is probably more challenging than SNLI. Apart from this, we observed some imprecisions in the dataset during manual inspection of some samples that our model was unable to predict correctly. Table 4 list three random samples from the test set which we believe contains ambiguous pair that are quite close-to-call. For instance, if Mr A works in a warehouse, we can safely assume that he's probably at the warehouse at a given time. However, this was labeled as a contradiction. Overall, we achieved a modest score of 74.2 % which can still be improved on by fine-tuning our model or optimizing some parameters.

4.3 Ablation and Error Analysis on Textual Entailment

Ablation test is usually performed to verify the importance, significance, or contribution of a component of a model or a particular feature to the overall performance of a system. Here, our goal is to see the part of our model that is most important to the performance of the model. Recall that our intra attention module focuses on the

[5] https://nlp.stanford.edu/projects/snli/

[6] https://inclass.kaggle.com/c/multinli-mismatched-evaluation/leaderboard

[7] As at the time of carrying out our experiment and initially writing this article, there was no available result on the dataset. We ran our experiment two days after the public release of the dataset

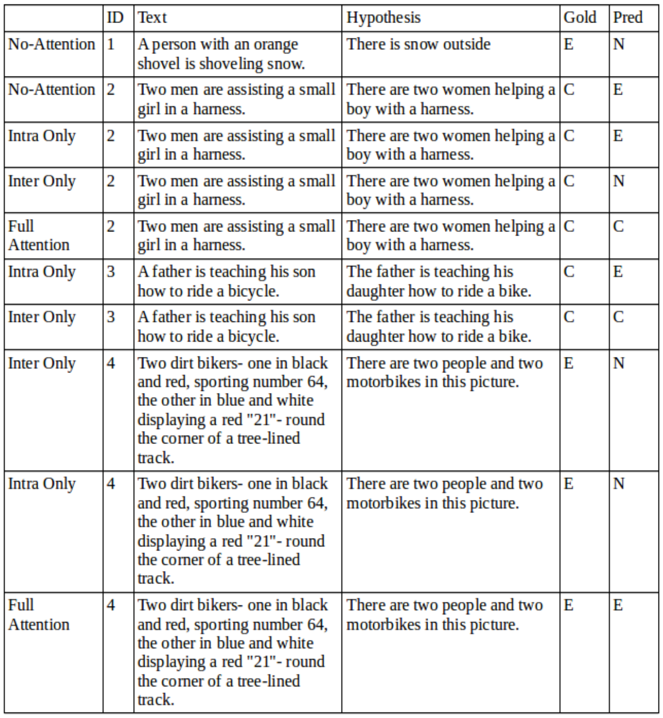
important words for each sentence while the inter attention module is our feature generation step that creates an interaction between one sentence representation to another sentence representation. In the first step, we removed the intra attention for both text and hypothesis models while retaining the inter-sentence attention. In the second experiment, we retained the intra-sentence attention while eliminating the inter-sentence attention steps. Apart from these settings, the remaining parts of the model were kept intact. Table 3 shows the results obtained under the two scenarios. Finally, we removed both intra and inter-sentence attention while also encoding words using a non-bi-directional LSTM. In this particular setting, we used a stacked LSTM to encode the words in each sentence and obtained a sentence representation for both text and hypothesis. Next, we utilized a form of distance function similar to the work described in [1]. This defaults to measuring the semantic similarity between the two text representations, and we specifically used the cosine formula also utilized in [32]. The result obtained is shown in Table 3 indicated as *No Attention*. The results obtained in the former ablation experiment is indicated as *Inter Attention without Intra Attention* and vice versa.

We see that the Inter-Sentence attention is more important to our model than the Intra-sentence attention. Furthermore, when Intra-sentence attention is used without Inter-sentence attention, our model still outperforms a number of systems. However, combining the two steps significantly improves the overall system performance. When we removed these important steps, (No Attention), the performance degrades badly, losing over 4% in classification performance. However, we see that the performances of our model under this ablation setup supersede those from [7, 16, 22] etc. Even though we did not replicate the ablation experiment across the tasks or datasets used in all the experiments reported in this work, relying on the significant difference in performance during ablation, there is every reason to believe that the behaviour of the individual component is truly reflected. We expect to arrive at similar results in other tasks like Answer Sentence Selection and Paraphrase Detection.

Figure 5 shows some cherry-picked text-hypothesis pairs form SNLI for error analysis. E, C and N stands for entailment, contradictory and Neutral relationship respectively. The Gold column contains the human assigned label while the pred contains the predicted label. We see that when no attention is used or where only intra-sentence attention is used and the sentences are long, the system struggles to predict the correct label. This can be noticed from example with ID #1. Also, the system tends to be identifying relatedness and not entailment when Intra attention is used in example #3. However, the system correctly predicts the same example when inter-sentence attention is in use. In example #4, the full attention module is able to associate 'two dirty bikers' to 'two people' and 'two motorbikes', while ignoring irrelevant information about colours from the text. However, the same example could not be correctly predicted when either of intra or inter sentence attention is used. As previously highlighted, even though the error analysis is not done across the tasks, we expect to see the same behavior in other tasks or datasets.

Models	**Accuracy**
Bowman et. al.,[6]	77.6
Rocktaschel et. al.,[22]	83.5
Liu et. al.,[16]	85.0
Shouohang et. al.,[30]	86.1
Cheng et. al.,[8]	86.3
No Attention	83.9
Intra Attention without Inter Attention	85.5
Inter Attention without Intra Attention	86.8
Full Attention	**88.2**

Table 3: Attention Ablation on SNLI dataset



	ID	Text	Hypothesis	Gold	Pred
No-Attention	1	A person with an orange shovel is shoveling snow.	There is snow outside	E	N
No-Attention	2	Two men are assisting a small girl in a harness.	There are two women helping a boy with a harness.	C	E
Intra Only	2	Two men are assisting a small girl in a harness.	There are two women helping a boy with a harness.	C	E
Inter Only	2	Two men are assisting a small girl in a harness.	There are two women helping a boy with a harness.	C	N
Full Attention	2	Two men are assisting a small girl in a harness.	There are two women helping a boy with a harness.	C	C
Intra Only	3	A father is teaching his son how to ride a bicycle.	The father is teaching his daughter how to ride a bike.	C	E
Inter Only	3	A father is teaching his son how to ride a bicycle.	The father is teaching his daughter how to ride a bike.	C	C
Inter Only	4	Two dirt bikers- one in black and red, sporting number 64, the other in blue and white displaying a red "21"- round the corner of a tree-lined track.	There are two people and two motorbikes in this picture.	E	N
Intra Only	4	Two dirt bikers- one in black and red, sporting number 64, the other in blue and white displaying a red "21"- round the corner of a tree-lined track.	There are two people and two motorbikes in this picture.	E	N
Full Attention	4	Two dirt bikers- one in black and red, sporting number 64, the other in blue and white displaying a red "21"- round the corner of a tree-lined track.	There are two people and two motorbikes in this picture.	E	E

Figure 5: Error analysis on SNLI dataset

4.4 Experiment on Paraphrase Detection

We utilized the Quora dataset for this part of our work. The Quora dataset is still a relatively new dataset which was only released in January, 2017. Thus, very few work already reports the performance of their system for this dataset. Nevertheless, we benchmarked our model's performance with some of the baselines reported by Wang et. al., [32]. Wang et. al., [32] already achieved a state-of-the-art performance in textual entailment with their system, thus making it a very strong baseline to compare with. Our result is provided in table 4. Interestingly, most of the baselines reported by [32] are also based on a kind of Siamese-architecture, however, their approach is strikingly different from ours. Moreover, we see that our model also outperforms all the baselines significantly. Even though the best performing system reported is to some order of magnitude better than ours, it is apparent that our model generalizes well enough. The second dataset for this task is the Microsoft Research Paraphrase Corpus (MSRP)[8]. We already analyzed this corpus in the previous section. Our model achieves a modest score of 82.0% on the test set while achieving a much higher result from the training set.

Models	**Quora**
Wang et. al., [32] s-CNN	79.60
Wang et. al., [32] m-CNN	81.38
Wang et. al., [32] s-LSTM	82.58
Wang et. al., [32] m-LSTM	83.21
Wang et. al., [32] BiMPM	**88.17**
This paper	83.6

Table 4: Paraphrase Detection result on Quora dataset

Test Split	**Accuracy**
Train	96.40
Validation	79.00
Test	82.00

Table 5: Paraphrase Detection result on MSR dataset

4.5 Experiment Answer Selection

The dataset for this task is the wikiCorpus which is an open domain question answering dataset that was extracted from the wikipedia. For each question, the authors selected Wikipedia pages and used sentences in the summary paragraph as candidates, which were then annotated on a crowdsourcing platform. We follow the same pre-processing steps that was carried out by [36], such that questions with no corresponding correct candidate answers are excluded. Table 6 reports the result obtained from the evaluation. We see that we achieved a better MRR scores than all but one system [30]. Also, our MAP score is better than all but two of the listed systems. Note that some of the benchmark systems are actually task specific unlike our model that is general purpose.

Models	**MAP**	**MRR**
Rao et. al., [21]	0.701	0.718
Yin et. al., [38]	0.692	0.711
Zhiguo et. al., [33]	0.706	0.723
He et. al., [11]	0.709	0.723
Santos et. al., [9]	0.689	0.696
Wang B. et. al., [29]	0.7341	0.7418
Shuohang et. al., [31]	**0.743**	**0.755**
Wang et. al., [32]	0.718	0.731
This paper	0.710	0.746

Table 6: Answer sentence selection result on WikiCorpus

5 CONCLUSION

Measuring similarity between two texts has a number of applications in natural language processing. In this paper, we have presented a Siamese architecture neural networks model which uses a type of recurrent neural networks called LSTM. Our main contribution is how we employed attention at different granularity of text, i.e., both at word level as well as at sentence level, which corresponds to the inter-sentence and intra-sentence attentions. The intra-sentence attention was used to measure the importance of each word within a sentence based on its similarity with a word-level context vector and consequently select the most important words to represent a sentence. Analogously, the inter-sentence attention was used to create an interaction between two sentences, by measuring the importance of the words in one sentence, conditioned on the words in the other sentence, i.e., using a sentence-level context vector. Consequently, we obtained two feature vectors which are high level representation for both the text and the hypothesis. We evaluated our system on three tasks, including textual entailment, paraphrase detection, and question answering. Even though we were unable to set a new state-of-the-art record, the evaluation shows that our model is competitive and as can be seen in the result tables, our model significantly outclassed several state-of-the-art baseline systems across the three semantic tasks that have been studied in this work. Our system has room for improvements, even if we perform a bit of task specific optimization which has not been carried out in this work. In the future, we would like to evaluate the significance of each component of our model separately in a comprehensive way. Specifically, we would like to see how the attention layers in our work impact performance and then provide both quantitative and qualitative analysis on why it performed creditably.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

Kolawole J. Adebayo has received funding from the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science and Technology.

Luigi Di Caro and Guido Boella have received funding from the European Union's H2020 research and innovation programme under the grant agreement No 690974 for the project "MIREL: MIning and REasoning with Legal texts".

REFERENCES

[1] Kolawole Adebayo, Luigi Di Caro, Livio Robaldo, and Guido Boella. 2016. Textual Inference with Deep Learning Technique. In *Proc. of the 28th Annual Benelux Conference on Artificial Intelligence (BNAIC2016).*

[2] E. Agirrea, C. Baneab, D. Cerd, M. Diabe, A. Gonzalez-Agirrea, R. Mihalceab, G. Rigaua, J. Wiebef, and B. Donostia. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. *Proceedings of SemEval* (2016), 497–511.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Petr Baudiš and Jan Šedivỳ. 2016. Sentence Pair Scoring: Towards Unified Framework for Text Comprehension. *arXiv preprint arXiv:1603.06127* (2016).

[5] Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Dang, and Danilo Giampiccolo. 2011. The seventh pascal recognizing textual entailment challenge. *Proceedings of TAC* 2011 (2011).

[6] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv*

[8] https://www.microsoft.com/en-us/download/details.aspx?id=52398

preprint arXiv:1508.05326 (2015).

[7] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038* (2016).

[8] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733* (2016).

[9] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR, abs/1602.03609* (2016).

[10] Minwei Feng, Bing Xiang, Michael R Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 813–820.

[11] Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of NAACL-HLT*. 937–948.

[12] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[13] Adebayo Kolawole John, Luigi Di Caro, and Guido Boella. 2016. NORMAS at SemEval-2016 Task 1: SEMSIM: A Multi-Feature Approach to Semantic Text Similarity. *Proceedings of SemEval* (2016).

[14] Chen Liu. 2013. *Probabilistic siamese network for learning representations*. Ph.D. Dissertation. University of Toronto.

[15] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Modelling Interaction of Sentence Pair with coupled-LSTMs. *arXiv preprint arXiv:1605.05573* (2016).

[16] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090* (2016).

[17] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model.. In *Interspeech*, Vol. 2. 3.

[18] Tsendsuren Munkhdalai and Hong Yu. 2016. Neural Tree Indexers for Text Understanding. *arXiv preprint arXiv:1607.04492* (2016).

[19] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* (2016).

[20] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–43.

[21] Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1913–1916.

[22] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).

[23] Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. [n. d.]. Reading and Thinking: Re-read LSTM Unit for Textual Entailment Recognition. ([n. d.]).

[24] R. Socher, E. Huang, J. Pennin, C. Manning, and A. Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*. 801–809.

[25] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 129–136.

[26] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[27] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[28] Ming Tan, Bing Xiang, and Bowen Zhou. 2015. LSTM-based Deep Learning Models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108* (2015).

[29] Bingning Wang, Kang Liu, and Jun Zhao. 2016. Inner attention based recurrent neural networks for answer selection. In *The Annual Meeting of the Association for Computational Linguistics*.

[30] Shuohang Wang and Jing Jiang. 2016. A Compare-Aggregate Model for Matching Text Sequences. *arXiv preprint arXiv:1611.01747* (2016).

[31] Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905* (2016).

[32] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. *arXiv preprint arXiv:1702.03814* (2017).

[33] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019* (2016).

[34] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698* (2015).

[35] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[36] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering.. In *EMNLP*. Citeseer, 2013–2018.

[37] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*. 1480–1489.

[38] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193* (2015).

[39] Xiang Zhang and Yann LeCun. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710* (2015).

[40] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. 649–657.