A study of models for Natural Language Inference (NLI) using the MultiNLI dataset

Author Student IDs: 20053948, 20129724, 20101171, 20150879

Abstract

This study is centered on the exploration and understanding of which parts of a Natural Language processing model, namely the encoding method, the vector merging blocks and the classifier have a large impact on the accuracy of the NLI classification as well how the attributes of these blocks impact prediction accuracy. The Bidirectional Encoder Representations from Transformers (BERT) based model and multiple encoder-clasifier combinations that have been developed were trained and tested for performing Natural Language Inference (NLI) task on the MultiNLI corpus. After performing several experiments to understand the sensitivity of the model performance to changes on each of the blocks in the architecture, the evaluation led to an insightful discussion about the impact of encoders, merging blocks and classifier on the model's accuracy.

1 Introduction

There is a wide variety of tasks within the field of Natural Language Understanding (NLU) for which technological models have been developed to process language better. One of the most useful tasks within this wide range of NLU tasks, is the task of Natural Language Inference (NLI). Natural Language Inference involves classifying sentence pairs to be related to one another in one of three ways; complimentary, contradictory or neutral. Complimentary sentence pairs have one sentence that entails the other. Neutral sentence pairs have sentences that are unrelated and contradictory sentence pairs have one sentence that contradicts the other.

Natural Language Inference (NLI) has become an important area of research for Natural Language Understanding (NLU) (Wang et al., 2018). Since the Stanford NLI (Williams et al., 2017) corpus appeared, different approaches have been developed for improving the accuracy of models that classify sentence pairs at the expense of the computational resource required (Reimers and Gurevych, 2019), sometimes making it impossible to be run on traditional computers. This was partially overcome by the use of the pre-trained models that are freely available in the community (E.g.: BERT).

Instead of looking for the most accurate model, this study is centered on the exploration and understanding of which part of the model most affects the accuracy for the NLI classification task. This will then be extended as recommendations for future work to orient the areas of focus for similar models and approaches.

All the model approaches for tasks aimed at investigating the relationship between two sentences mainly have two structures,that are shown in figure 1.They are the representation based model and the interaction based model respectively. The latter takes both sentences as input at the same time, while the representation based model processes each sentence separately. For this study, the representation based model is used as it is possible to analyze and modify each of the separate blocks in this architecture. Furthermore, under this approach, the following specific research questions were proposed and later answered to reach the conclusion for the core purpose. This was the main aim of the paper and the questions were as follow: (i)whether the encoding method has a large impact on accuracy and what makes a good encoder?; (ii) which sets of vector merging blocks provide the most information, if they do at all? (iii) which classifier, if any leads to the best prediction accuracy?

Section 2 of this report describes the related works in detail. Section 3 and 4 aim to answer the first specific question. Section 3 explains not only several different encoders which are LSTM, BiLSTM, BiGRU and self-attention but also explains the pre-trained BERT model encoder with a simple classifier. Section 4 experiments these

models with the fixed classifiers in a given data set. In addition, the same training details for the entire experiments are also included in this section. Section 5 aims to answer the second and the third specific questions. It tests the effect of several different classifiers using softmax with BiLSTM and self-attention as encoders. It is assumed that with BiLSTM, LSTM and BiGRU, the influence of these encoders is similar due to their similar structures. Finally, combining the answers to these specific questions, the final conclusion as well as suggestions for further investigations were proposed.

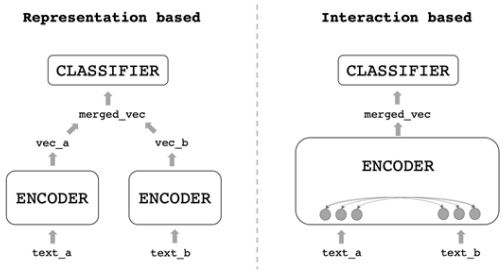

Figure 1: Relationship model for sentence comparison

2 Related Works

A set of 5 papers were reviewed to understand the relevant information in the field of Natural Language Understanding, with a particular focus on NLI, encoders and embeddings.

The paper "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference" was reviewed first (Williams et al., 2017). This paper introduced and discussed the relevance of the MultiNLI corpus to Natural Language Inference that our team used to train our models. In the paper, 433 thousand sentence pair examples were introduced for natural language inference (a.k.a. recognizing textual entailment; which is the objective of the models used in our report). The paper made a significant contribution to the field as it improved upon the available resources in terms of both coverage and difficulty. The Stanford NLI Corpus (SNLI) was a previous benchmark that was initially used for NLU. However the sentences in this previous corpus were derived from only text genre-image captions and were limited to descriptions of concrete visual scenes. The MultiNLI corpus was suggested to be far better as it contained sentence data from ten distinct genres of written and spoken English making it far more effective in terms of coverage and thus making it possible to evaluate systems on nearly the full complexity of the English Language. It also contained a much higher percentage of sentences tagged with one or more elements from a set of thirteen difficult linguistic phenomena. To verify this, three neural network models were used with the SNLI corpus and MultiNLI corpus. The model prediction accuracy and the difficulty of the data sets were evaluated and it was found that the MultiNLI corpus was a more difficult corpus but also a more comprehensive one. It was the one that we used for our model evaluation.

The second paper reviewed was titled "Natural Language Inference Over Interaction Space" (Gong et al., 2017). This paper focused on performing natural language inference by introducing a novel class of neural network architectures known as Interactive Inference Network (IIN), to achieve high-level understanding of the sentence pairs. It does this by hierarchically extracting semantic features from interaction space. It was stated to be the first known attempt to solve an NLI task in the interaction space.

The experiments done in the paper demonstrated that by capturing rich semantic features in an interaction tensor (attention weight) with the IIN, NLI tasks had very good outcomes in terms of performance, especially in cases with paraphrase, antonyms and overlapping words. This interaction tensor is what is created in an interaction based NLP model. The relevance of this paper is that one instance of the IIN architecture, namely the Densely Interactive Inference Network (DIIN (ensemble)) model achieved a test matched accuracy of 80 percent (the highest value amongst the many IIN models) on the same MultiNLI sentence pair corpus we used. Since our team was able to reach a test matched accuracy above this using a BERT representation based model on the MultiNLI data set, we believe it is a valuable piece of work when compared to what was done in the past papers despite it being a representation based.

The third paper reviewed was titled "Learning General Purpose Distributed Sentence Representations Via Large Scale Multi-Task Learning" (Subramanian et al., 2018). Prior to this paper, research was focused on distributed embedded vector representations of words trained

on large amounts of text in an unsupervised manner. These representations were typically used as general purpose features for words in many NLP tasks. However the novelty of this paper is that it delves into unsupervised and supervised learning techniques to learn general purpose fixed-length sentence representations instead of word representations. The rationale behind this research was that these general purpose sentence representations would be beneficial in computationally resource scarce environments. Another interesting aspect is that the model used in this paper was trained on over a 100 million sentences. The relevance of this paper is that of the 6 tasks (multi-task training) that this model was trained on, one of them was the NLI task that we performed. It was done with 1 million training sample sentence pairs from the SNLI and MultiNLI corpus. The objective of this paper was met in that it showed that with extensive experiments that share a single recurrent sentence encoder across weakly related tasks, consistent improvements over previous methods to provide general purpose sentence representations, can be achieved.

The fourth paper reviewed was titled "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks" (Reimers and Gurevych, 2019). The objective of this paper was to reduce the computational time of BERT to determine textual semantic similarity. For example, finding the most similar pair in a collection of 10000 sentences requires about 50 million inference computations which takes about 65 hours with BERT. This long duration makes BERT unsuitable for semantic similarity search and for unsupervised tasks like clustering. The performance of BERT for seven Semantic Text Similarity (STS) tasks is also below the performance of average GloVe embeddings (discussed in section 3). This paper thus presented a better solution to reduce the computational time and improve performance on STS tasks via a new method called Sentence-BERT (SBERT). This SBERT model is a modification of the original pretrained BERT network that uses Siamese and triplet network structures to derive semantically meaningful (sentences with embeddings that are closest to each other in vector space) sentence embeddings with cosine-similarity. This lead to the outcome of reducing the time for finding the most similar sentence pair from 65 hours with BERT to about 5 seconds with SBERT, without a loss of accuracy. SBERT adds a pooling operation to the output of BERT to derive a fixed sized sentence embedding. The figure below shows an example of how this model works.

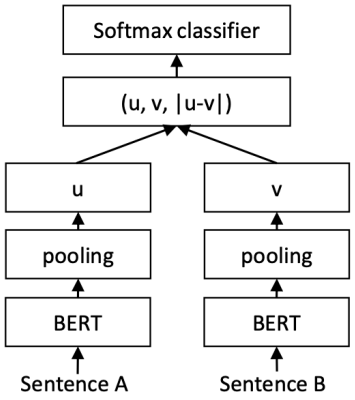

Figure 2: SBERT architecture (Reimers and Gurevych, 2019)

SBERT was trained on a combination of SNLI and MultiNLI corpus datasets with the default mean pooling strategy. The SBERT sentence embeddings were compared to other sentence embeddings methods on the seven well known sentence evaluation transfer tasks. It was found that SBERT had the best performance in 5 out of the 7 tasks. Thus it was found that not only could SBERT reduce the computational time to derive textual semantic similarity, it also could achieve a significant improvement over the existing state of the art sentence embeddings methods for the seven commonly used Sentence Evaluation tasks for benchmarking.

The final paper reviewed was titled "Glue: A Multi-Task Benchmark And Analysis Platform For Natural Language Understanding" (Wang et al., 2018). This paper was novel as it introduced a benchmark known as the General Language Understanding Evaluation (GLUE) benchmark that is a collection of tools for evaluating the performance of models across a diverse set of existing Natural Language Understanding (NLU) tasks. The goal with this benchmark was to improve NLU technology to process language in a way that is not exclusive to any task, genre or dataset.

Through the analysis, it was found that, in aggre-

gate, models trained jointly on the several GLUE tasks outlined had better performance than the combined performance of models trained for each task separately.

One of the tasks in this paper that was included in the benchmark of NLU tasks was the NLI task that we performed with our models. However, the paper served primarily as a means to understand the range of NLU tasks that can serve as a good benchmark when evaluating a Natural Language Processing model. Nine English sentence understanding tasks, which cover a broad range of domains, data quantities, and difficulties were done. Two of them were single-sentence tasks. Three of them were similarity and paraphrase tasks and the remaining four were inference tasks.

3 Methods

3.1 Research Framework

The model used in this study for the NLI task is based on the representation based model scheme shown in the figure 1. To explore the specific research question, different combinations of word embedding, encoders, vector merging blocks and classifiers were tested. Figure 3 summarizes the combinations tested as a representation based scheme. The scheme is composed of two Siamese encoders to produce an embedded representation vector for each sentence u,v. These two encoders share the same training parameters. Before feeding into the classifier, the two output sentence embedding vectors are merged into a single vector in the vector merging block of the architecture. Finally, a classifier in the form of a neural network (with activation functions) classifies the relationship between the two sentences. In the vector merging block part, the multiple operations such as concatenation, subtraction and multiplication were applied to the embedded sentence vectors. This adds more information to the vector to be used afterwards by the classifier. For instance, the multiplication of the vectors is related to the cosine similarity, which is an important metric for measuring similarity between embeddings.

In the classifier block of the architecture, different linear neural networks with one and two hidden layers and with or without the activation function $tanh(x)$ were used. This is common for classification tasks. In the next subsections, each of the encoders used, namely LSTM, InferSent, BiGRU, self-attention and BERT, are explained in detail.

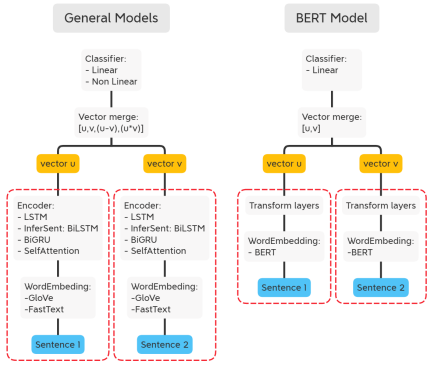

Figure 3: The Generic NLI training scheme

3.2 Encoders

In the following subsections, the different encoders considered are described.

3.2.1 Word Embedding

With the exception of BERT, all the encoders used in this study generated a sentence embedding representation from the word embedding input representation. Input sentences were first transformed using a pre-trained word embedding algorithm such as GloVe (Pennington et al., 2014) and FastText (Mikolov et al., 2017).

3.2.2 LSTM

The first encoder considered in this study is the Long Short-Term Memory Neural Network (LSTM) (Schmidhuber and Hochreiter, 1997). LSTM has a Recurrent Neural Network (RNN) architecture with gates that allow feedback to generate a sequence of T hidden states representations $h_1, \ldots, h_T$ where $h_t = \overrightarrow{LSTM}(w_1, \ldots, w_T)$. With these gates, LSTM avoids exploding and vanishing gradients problem that normal RNNs are exposed to. Figure 4 below represents the differences between the RNN and LSTM "neuron" architectures.

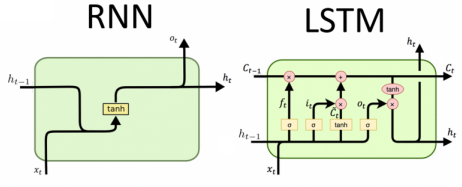

Figure 4: RNN and LSTM architectures (KaliaFollow-Software)

3.2.3 InferSent: BiLSTM max-pooling

Figure 5 shows an InferSent model, which is formed by a BiLSTM model with max-pooling (Collobert and Weston, 2008). BiLSTM consists of two LSTM encoder models: one LSTM in the forward direction and another LSTM in the backwards direction. This bidirectional representation of the sentence means that a sentence can be represented by each of the LSTM models in the forward and backward directions respectively. This improvement solves issues around the higher importance given by single LSTM models to the last terms, in comparison to the first term and also adds more contextual information to the model. Mathematically, T hidden states are generated: $\overleftrightarrow{h}_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]$ with $\overrightarrow{h}_t = \overrightarrow{LSTM}(w_1, \ldots, w_T)$ and $\overleftarrow{h}_t = \overleftarrow{LSTM}(w_1, \ldots, w_T)$.

Finally, the output representation is the max-pooling of these hidden states, i.e $u = max(\overleftrightarrow{h}_1, \ldots, \overleftrightarrow{h}_T)$

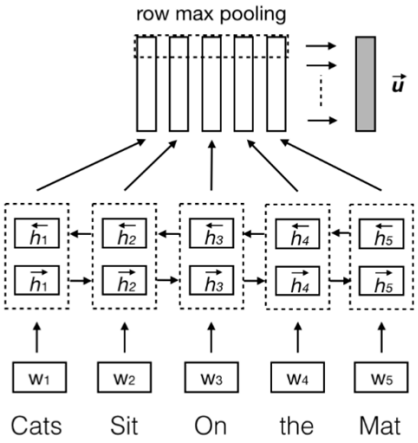

Figure 5: Bi-LSTM max-pooling network (Lin et al., 2017)

3.2.4 BiGRU

Gated Recurrent Unit (GRU) models (Cho et al., 2014), are simpler LSTM models with fewer gates, making them more computationally efficient than LSTM models. It is also possible to train Bidirectional GRU models (BiGRU) that create representations in two directions. Finally, the output representation is the concatenation of the last hidden states of the forward and backward GRU as shown on Figure 6.

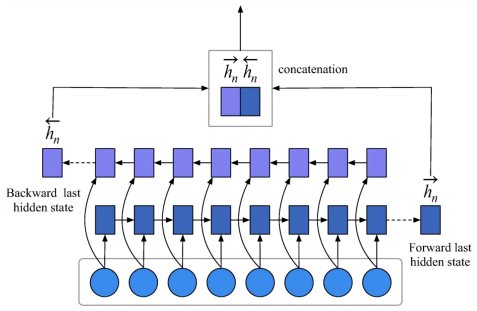

Figure 6: Last hidden state BiGRU architecture (Zhang et al., 2019)

3.2.5 Self-attention

Self-attention models used in this study are based on the attention based hierarchical LSTM (Wang and Fan, 2018b) which is shown on figure 7. The self-attention mechanism takes the BiLSTM idea even further by concatenating the hidden states and the BiLSTM outputs to generate the embedding representation of the next word. This allows the model to create the embedding by focusing on certain parts of the input. The attention mechanism used in this study is defined by the following equations:

$$\bar{h}_i = tanh(Wh_i + b_w)$$

$$\alpha_i = \frac{e^{\bar{h}_i^T u_w}}{\sum_i e^{\bar{h}_i^T u_w}}$$

$$u = \sum_t \alpha_i h_i.$$

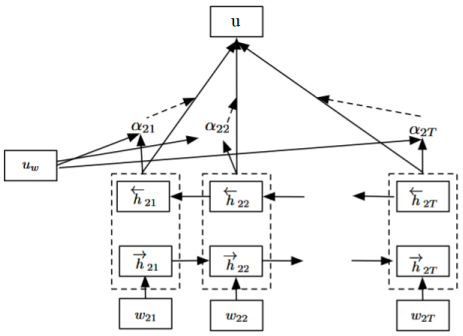

Figure 7: Inner Attention word level architecture (Wang and Fan, 2018a)

3.2.6 BERT

Transformer architecture The Transformer is an advanced model architecture (shown in Figure 8 below) based on self-attention mechanism (Vaswani et al., 2017). This model can extract features for each word to find out the importance of all other words in the sentence. It is very parallel and

efficient, as no recurrent units are used, and only weighted sums and activations are used.

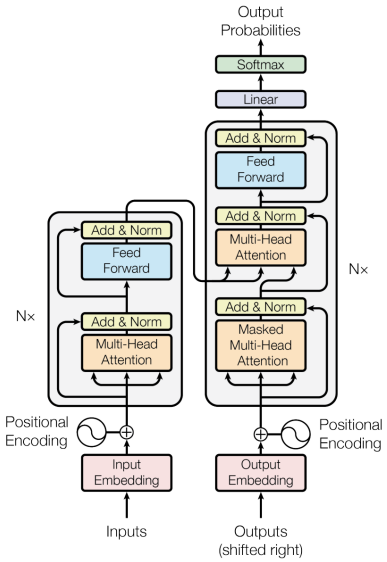

Figure 8: Model architecture of the Transformer (Vaswani et al., 2017)

Fine-tuning We used the 'BERT-base-uncased' version for this task, which was first introduced in this paper (Reimers and Gurevych, 2019). This model was pretrained for English on the Wikipedia and BooksCorpus in a self-supervised way, and the inputs were "uncased", which indicates that the text was lower-cased before tokenization. The model contains 12 Transformer layers, and a hidden size of H=768. Each layer receives the token embedding list and produces the same number of embeddings on the output. To fine-tune the model, we added one classification layer which contains 3 output to the end of 12 Transformer layers.

4 Experiments and Results About Encoder

4.1 Experiments

Data The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information[1]. This was the main data set used for training and testing in the models investigated.

Optimizer Adam optimizer with weight decay (AdamW) learning rate 10^{-5} was used in BERT fine-tuning. This optimizer computes adaptive learning rates per parameter, and estimates the first and second moments of the gradient, in addition the weight decay regularization is employed (Loshchilov and Hutter, 2019). For the other encoders, SGD (Stochastic Gradient Descent) was used instead of AdamW due to memory issues in the these. The initial learning rate for the model was set to be 0.5 with a 5% decay rate in each iteration. In addition, instead of gradient clipping, the learning rate was divided by 5 when the accuracy in the validation set began to drop. When the learning rate decreased to a value under 10^{-5} or the iteration number reached 20, the model stopped training.

Loss There are three label classes in our data, and they are provided as integers. To compute the loss, the sparse categorical cross entropy was employed. It computes the cross entropy between the predicted values and the labels.

Metrics The classification accuracy was provided, which is the fraction of predictions our model got right. The definition by function is:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Batch size Batch size refers to the number of input data points in each mini-batch. In this case, we used batch sizes of 36 for both models, which is recommended by BERT authors (Devlin et al., 2018).

4.2 Results and Discussion

The model was trained on the MultiNLI corpus dataset where *matched* and *mismatched* are treated as the validation and test dataset respectively. Words in the dataset are embedded by the pretrained model. The text was made into lowercase letters without performing stemming or removing stop words, that concurs with best practices (Henkel, 2018).

The results show that the performance of BERT is significantly better than other models in every way, since its accuracy is around 15% larger than others. It can be noticed that the accuracy with BiLSTM and attention as an encoder in the validation/test dataset are quite close. However, this is not enough to determine whether the difference in the accuracy is significant or not.

To determine if this difference is significant, a statistical test was implemented. By modelling

[1] https://cims.nyu.edu/~sbowman/multinli/

Model	*matched*	*mismatched*
LSTM	63.25	64.06
BiLSTM	69.11	69.02
BiLSTM(FastText)	67.51	67.98
BiGRU	65.36	66.54
Self-attention	69.44	69.01
BERT	83.29	83.30

Table 1: Models performance on MultiNLI task test sets. The values are scaled by 100.

whether the model could correctly classify the relationship of the two sentences as following Bernoulli distribution (1 for true and 0 for false). Then, the mean value of it (accuracy) would follow the normal distribution according to the central limit theorem, and as the deviance for the population is unknown and the measurements were taken from the same "individual" (pair of sentences), a paired t-test should be applied, with the null hypothesis that the accuracy of the compared models are equal. It was found, that the p-value for rejecting the null hypothesis for the BiLSTM and attention models were 0.4701 and 0.9821 in the valid and test dataset respectvely. Thus, there is not enough evidence to reject the hypothesis. However, for the BiLSTM with the BiGRU model, the p-value in validation and test datasets are 2.42×10^{-5} and 5.47×10^{-8} respectively. Therefore, from this two-tailed test, it can be inferred that BiLSTM performs "better" than BiGRU.

Based on the hypothesis tests, it can be said that BERT models are more accurate than BiLSTM and self-attention models which are in turn more accurate than LSTM and BiGRU models. This suggests a positive relation between the accuracy achieved by the model and the quality of the information from the embedding produced by the encoders that is given to the classifier. BERT uses an advanced self-attention-based model Transformer. More information between each word with each others can be extracted compare with other models, which leads to a better performance. In the same way, BiLSTM and Self Attention encoders provide more information than LSTM as they have a double LSTM direction architecture. Furthermore, while BiGRU has a bidirectional architecture, its output embedding representation is the concatenation of only the last encoders in each direction, causing a loss of information related to words in the middle of the sentences.

An experiment that led to a change in the word embedding from GloVe to FastText within BiLSTM was also done. This test is reflected in Table 1 and shows that the model using GloVe as a word embedding behaves better (statistically significant). The reason is likely that GloVe can leverage global statistical information contained in the sentences, which may be quite useful in this task.

Additionally, the result of the BERT models ws analysed using the confusion matrix (shown in figure 9). The matrix is normalized, by dividing the entries by the total number of true labels in each classified group. It shows that the model tends to misclassify the entailment into neutral rather than into contradiction or misclassifies the contradiction into neutral rather than into entailment. As the softmax function cannot distinguish the order information, it suggests there may be ordinal information with regard to language inference inside the encoder. To extract this information should be the key point to make a good encoder. This feature also exists in other encoders, which was omitted here.

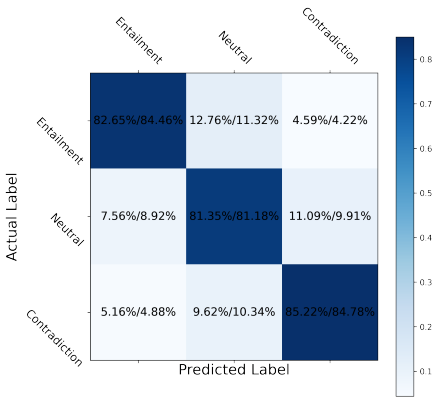

Figure 9: BERT Confusion Matrix

5 Merging vector and Classifier

As mentioned in the introduction, for models with the BiLSTM and self-attention encoder, the concatenated vector and the classifier using the softmax framework were changed to test the their influence. The results are shown in table 2.

1. The accuracy decreases dramatically by changing the concatenated vector from $(u, v, u * v, |u - v|)$ to just (u, v), and it is statistically significant. This may be caused by the decrease of the encoder dimension (size of the output sentence embedded vector), which

Model	*matched*	*mismatched*
Original model	69.1/69.4	69.0/69.0
Half encoding-dim	68.1/69.7	68.7/69.6
Delete $u * v \& \lvert u - v \rvert$	56.6/57.4	56.7/57.7
Non-linear layer	68.1/70.0	68.9/69.8
-1 linear layer	68.6/69.8	69.4/69.2
+1 non-linear layer	67.7/70.4	68.6/69.6
Double layer dimension	69.1/69.5	69.0/69.0

Table 2: Models performance on MultiNLI task development sets. The two values in each entry refer to the accuracy (scaled by 100) from the BiLSTM model and the self-attention model. respectively

is proven to have an effect on the accuracy in (Wang and Fan, 2018a). Nevertheless, the results show that in comparison with the original model, the model with half the encoding dimension size is more accurate than the model with the concatenated vector of just (u, v). This suggests that there is more meaningful information kept by the operation of calculating the absolute difference and multiplying the two embedded sentence vectors than doubling the encoding dimension. Thus, the merging vector block is relevant. Additionally, the model with fewer encoding dimensions, and that considers the three merging operations, is computationally cheaper as the encoder size is smaller. This reduces the training time and the GPU usage.

2. Change the linear layers between the fully-connected layer and the final 3 classes layer into a non-linear layer by adding the activation function $tanh$ and adopt dropout to avoid overfitting. After implementation, it was however found that the accuracy did not increase. One plausible reason might be related to the vanishing gradient associated with $tanh$ activation as most values in the embedding representation are already in the interval [0,1]. Thus, non-linearities are causing a training problem.

3. Delete one linear hidden layer or add one non-linear layer. To measure the effect of the two hidden linear layers, one of them was deleted, resulting in the increased accuracy of the test set (not statistically significant). To measure the effect of a deeper layer, a non-linear layer was added within the non-linear layer sets described earlier. The reason why a linear layer was not added to the hidden linear sets is because multiple series of linear layer combinations is equivalent to one here; this can be be proven mathematically. From the results, it seems that the accuracy in test set drops about 0.3% (not statistically significant).

4. Increase the neuron units in the linear layer to be double its current value (thus growing to 512). However, even when the linear layer was made twice as wide, the accuracy did not increase further.

6 Conclusions

The study centred on the exploration and understanding of which block of the representation based model most affects the accuracy for NLI classification. These results could be used to provide robust recommendations for model architecture decisions in future work.

In this paper, the different representation based models were trained and tested on the MultiNLI corpus of 433k sentence pairs. Using ceteris paribus analysis, it was found that different encoders (LSTM, BiLSTM, BiGRU, self-attention and BERT) result in different prediction accuracy values that is statistically significant. This suggests a positive relationship between the accuracy achieved and the quality of the information generated by the encoder. In terms of the merging of the encoding vectors, the result showed that the multiplication of the encoded sentence vectors and their absolute difference can express more information about sentences and thus improve the prediction accuracy. Finally, with regard to the classifier, multiple combinations were tested without observing any statistically significant difference.

All in all, the findings of this work can be summarised in the following statements. The encoder which aims at extracting features is the most important. A good representation of the encoding and its merger can save computational time, while ensuring higher accuracy. Finally, the classifier structure does not require further investigation.

The limitation of the report is that the findings may only be suitable for similar models. In addition, the interaction among blocks are considered during experiment design but not investigated, which is why further work with these models is recommended.

References

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Christof Henkel. 2018. How to: Preprocessing when using embeddings.

Robin KaliaFollowSoftware. Recurrent neural networks (rnn), gated recurrent units (gru), and long short-term memory (lstm).

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. international conference on learning representations (iclr). *International Conference on Learning Representations (ICLR)*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

B. Wang and B. Fan. 2018a. Attention-based hierarchical lstm model for document sentiment classification. *IOP Conference Series: Materials Science and Engineering*, 435:012051.

Bo Wang and Binwen Fan. 2018b. Attention-based hierarchical LSTM model for document sentiment classification. *IOP Conference Series: Materials Science and Engineering*, 435:012051.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Yuteng Zhang, Wenpeng Lu, Weihua Ou, Guoqiang Zhang, Xu Zhang, Jinyong Cheng, and Weiyu Zhang. 2019. Chinese medical question answer selection via hybrid models based on cnn and gru. *Multimedia Tools and Applications*, 79(21-22):14751–14776.