Learning Semantic Sentence Embeddings using Pair-wise Discriminator

Badri N. Patro* **Vinod K. Kurmi*** **Sandeep Kumar*** **Vinay P. Namboodiri**
Indian Institute of Technology, Kanpur
{badri,vinodkk,sandepkr,vinaypn}@iitk.ac.in

Abstract

In this paper, we propose a method for obtaining sentence-level embeddings. While the problem of securing word-level embeddings is very well studied, we propose a novel method for obtaining sentence-level embeddings. This is obtained by a simple method in the context of solving the paraphrase generation task. If we use a sequential encoder-decoder model for generating paraphrase, we would like the generated paraphrase to be semantically close to the original sentence. One way to ensure this is by adding constraints for true paraphrase embeddings to be close and unrelated paraphrase candidate sentence embeddings to be far. This is ensured by using a sequential pair-wise discriminator that shares weights with the encoder that is trained with a suitable loss function. Our loss function penalizes paraphrase sentence embedding distances from being too large. This loss is used in combination with a sequential encoder-decoder network. We also validated our method by evaluating the obtained embeddings for a sentiment analysis task. The proposed method results in semantic embeddings and outperforms the state-of-the-art on the paraphrase generation and sentiment analysis task on standard datasets. These results are also shown to be statistically significant.

1 Introduction

The problem of obtaining a semantic embedding for a sentence that ensures that the related sentences are closer and unrelated sentences are farther lies at the core of understanding languages. This would be relevant for a wide variety of machine reading comprehension and related tasks such as sentiment analysis. Towards this problem, we propose a supervised method that uses a sequential encoder-decoder framework for paraphrase generation. The task of generating paraphrases is closely related to the task of obtaining semantic sentence embeddings. In our approach, we aim to ensure that the generated paraphrase embedding should be close to the true corresponding sentence and far from unrelated sentences. The embeddings so obtained help us to obtain state-of-the-art results for paraphrase generation task.

Our model consists of a sequential encoder-decoder that is further trained using a pairwise discriminator. The encoder-decoder architecture has been widely used for machine translation and machine comprehension tasks. In general, the model ensures a 'local' loss that is incurred for each recurrent unit cell. It only ensures that a particular word token is present at an appropriate place. This, however, does not imply that the whole sentence is correctly generated. To ensure that the whole sentence is correctly encoded, we make further use of a pair-wise discriminator that encodes the whole sentence and obtains an embedding for it. We further ensure that this is close to the desired ground-truth embeddings while being far from other (sentences in the corpus) embeddings. This model thus provides a 'global' loss that ensures the sentence embedding as a whole is close to other semantically related sentence embeddings. This is illustrated in Figure 1. We further evaluate the validity of the sentence embeddings by using them for the task of sentiment analysis. We observe that the proposed sentence embeddings result in state-of-the-art performance for both these tasks.

* Equal contribution

Our contributions are: a) We propose a model for obtaining sentence embeddings for solving the paraphrase generation task using a pair-wise discriminator loss added to an encoder-decoder network. b) We show that these embeddings can also be used for the sentiment analysis task. c) We validate the model using standard datasets with a detailed comparison with state-of-the-art methods and also ensure that the results are statistically significant.

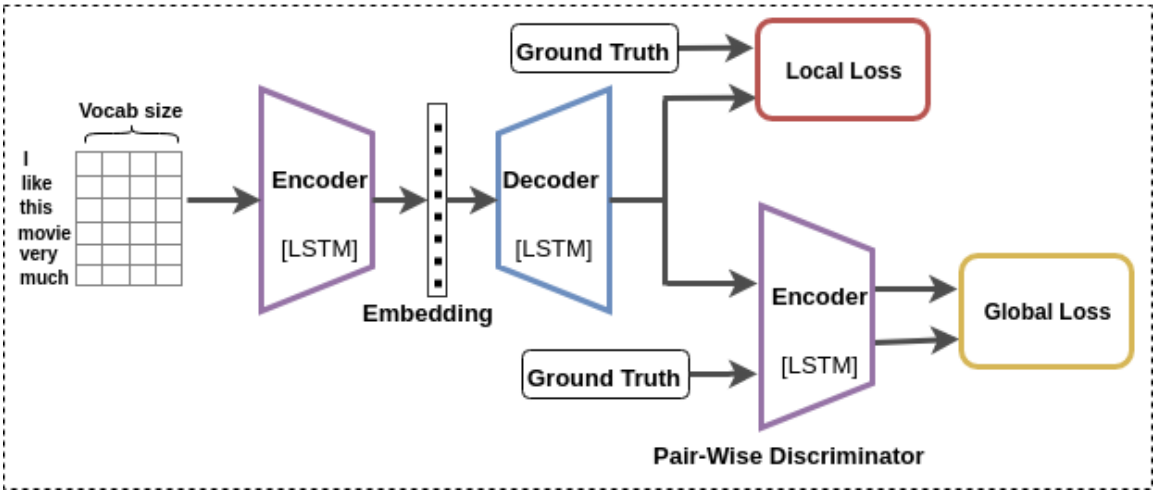

Figure 1: Pairwise Discriminator based Encoder-Decoder for Paraphrase Generation: This is the basic outline of our model which consists of an LSTM encoder, decoder and discriminator. Here the encoders share the weights. The discriminator generates discriminative embeddings for the Ground Truth-Generated paraphrase pair with the help of 'global' loss. Our model is jointly trained with the help of a 'local' and 'global' loss which we describe in section 3.

2 Related Work

Given the flexibility and diversity of natural language, it has always been a challenging task to represent text efficiently. There have been several hypotheses proposed for representing the same. (Harris, 1954; Firth, 1957; Sahlgren, 2008) proposed a distribution hypothesis to represent words, i.e., words which occur in the same context have similar meanings. One popular hypothesis is the bag-of-words (BOW) or Vector Space Model (Salton et al., 1975), in which a text (such as a sentence or a document) is represented as the bag (multiset) of its words. (Lin and Pantel, 2001) proposed an extended distributional hypothesis and (Deerwester et al., 1990; Turney and Littman, 2003) proposed a latent relation hypothesis, in which a pair of words that co-occur in similar patterns tend to have similar semantic relation. Word2Vec(Mikolov et al., 2013a; Mikolov et al., 2013b; Goldberg and Levy, 2014) is also a popular method for representing every unique word in the corpus in a vector space. Here, the embedding of every word is predicted based on its context (surrounding words). NLP researchers have also proposed phrase-level and sentence-level representations (Mitchell and Lapata, 2010; Zanzotto et al., 2010; Yessenalina and Cardie, 2011; Grefenstette et al., 2013; Mikolov et al., 2013b). (Socher et al., 2011; Kim, 2014; Lin et al., 2015; Yin et al., 2015; Kalchbrenner et al., 2014) have analyzed several approaches to represent sentences and phrases by a weighted average of all the words in the sentence, combining the word vectors in an order given by a parse tree of a sentence and by using matrix-vector operations. The major issue with BOW models and weighted averaging of word vectors is the loss of semantic meaning of the words, the parse tree approaches can only work for sentences because of its dependence on sentence parsing mechanism. (Socher et al., 2013; Le and Mikolov, 2014) proposed a method to obtain a vector representation for paragraphs and use it to for some text-understanding problems like sentiment analysis and information retrieval.

Many language models have been proposed for obtaining better text embeddings in Machine Translation (Sutskever et al., 2014; Cho et al., 2014; Vinyals and Le, 2015; Wu et al., 2016), question generation (Du et al., 2017), dialogue generation (Shang et al., 2015; Li et al., 2016b; Li et al., 2017a),

document summarization (Rush et al., 2015), text generation (Zhang et al., 2017; Hu et al., 2017; Yu et al., 2017; Guo et al., 2017; Liang et al., 2017; Reed et al., 2016) and question answering (Yin et al., 2016; Miao et al., 2016). For paraphrase generation task, (Prakash et al., 2016) have generated paraphrases using stacked residual LSTM based network. (Hasan et al., 2016) proposed a encoder-decoder framework for this task. (Gupta et al., 2017) explored a VAE approach to generate paraphrase sentences using recurrent neural networks. (Li et al., 2017b) used reinforcement learning for paraphrase generation task.

3 Method

In this paper, we propose a text representation method for sentences based on an encoder-decoder framework using a pairwise discriminator for paraphrase generation and then fine tune these embeddings for sentiment analysis task. Our model is an extension of *seq2seq* (Sutskever et al., 2014) model for learning better text embeddings.

3.1 Overview

Task: In the paraphrase generation problem, given an input sequence of words $X = [x_1, ..., x_L]$, we need to generate another output sequence of words $Y = [q_1, ..., q_T]$ that has the same meaning as X. Here L and T are not fixed constants. Our training data consists of M pairs of paraphrases $\{(X_i, Y_i)\}_{i=1}^{M}$ where X_i and Y_i are the paraphrase of each other.

Our method consists of three modules as illustrated in Figure 2: first is a Text Encoder which consists of LSTM layers, second is LSTM-based Text Decoder and last one is an LSTM-based Discriminator module. These are shown respectively in part 1, 2, 3 of Figure 2. Our network with all three parts is trained end-to-end. The weight parameters of encoder and discriminator modules are shared. Instead of taking a separate discriminator, we shared it with the encoder so that it learns the embedding based on the 'global' as well as 'local' loss. After training, at test time we used encoder to generate feature maps and pass it to the decoder for generating paraphrases. These text embeddings can be further used for other NLP tasks such as sentiment analysis.

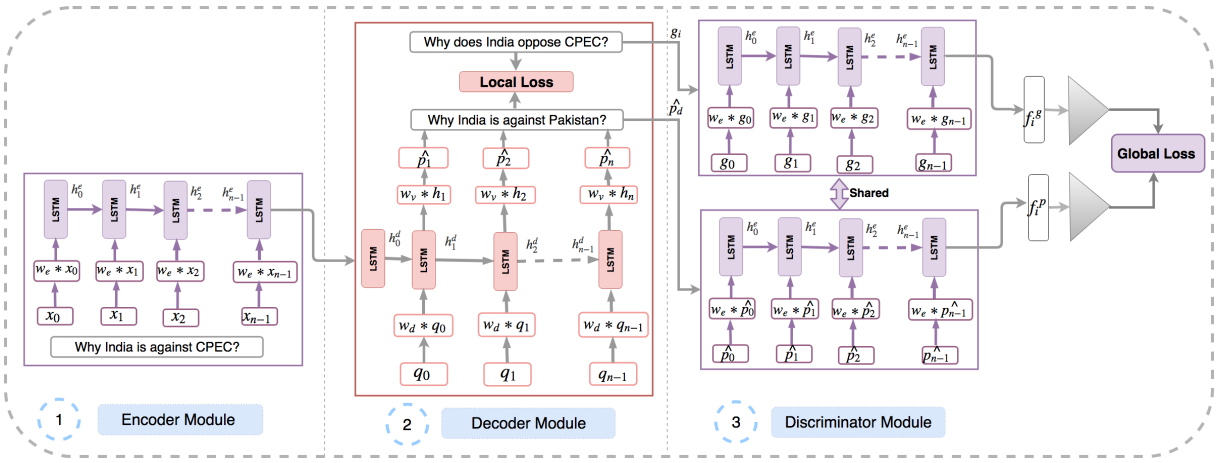

Figure 2: This is an overview of our model. It consists of 3 parts: 1) LSTM-based Encoder module which encodes a given sentence, 2) LSTM-based Decoder Module which generates natural language paraphrases from the encoded embeddings and 3) LSTM-based pairwise Discriminator module which shares its weights with the Encoder module and this whole network is trained with local and global loss.

3.2 Encoder-LSTM

We use an LSTM-based encoder to obtain a representation for the input question X_i, which is represented as a matrix in which every row corresponds to the vector representation of each word. We use a one-hot vector representation for every word and obtain a word embedding c_i for each word using a Temporal

CNN (Zhang et al., 2015; Palangi et al., 2016) module that we parameterize through a function $G(X_i, W_e)$ where W_e are the weights of the temporal CNN. Now this word embedding is fed to an LSTM-based encoder which provides encoding features of the sentence. We use LSTM (Hochreiter and Schmidhuber, 1997) due to its capability of capturing long term memory (Palangi et al., 2016). As the words are propagated through the network, the network collects more and more semantic information about the sentence. When the network reaches the last word (L_{th} word), the hidden state h_L of the network provides a semantic representation of the whole sentence conditioned on all the previously generated words $(q_0, q_1..., q_t)$. Question sentence encoding feature f_i is obtained after passing through an LSTM which is parameterized using the function $F(C_i, W_l)$ where W_l are the weights of the LSTM. This is illustrated in part 1 of Figure 2.

3.3 Decoder-LSTM

The role of decoder is to predict the probability for a whole sentence, given the embedding of input sentence (f_i). RNN provides a nice way to condition on previous state value using a fixed length hidden vector. The conditional probability of a sentence token at a particular time step is modeled using an LSTM as used in machine translation (Sutskever et al., 2014). At time step t, the conditional probability is denoted by $P(q_t|f_i, q_0, .., q_{t-1}) = P(q_t|f_i, h_t)$, where h_t is the hidden state of the LSTM cell at time step t. h_t is conditioned on all the previously generated words $(q_0, q_1.., q_{t-1})$ and q_t is the next generated word.

Generated question sentence feature $\hat{p_d} = \{\hat{p_1}, \ldots, \hat{p_T}\}$ is obtained by decoder LSTM which is parameterized using the function $D(f_i, W_{dl})$ where W_{dl} are the weights of the decoder LSTM. The output of the word with maximum probability in decoder LSTM cell at step k is input to the LSTM cell at step $k+1$ as shown in Figure 2. At $t = -1$, we are feeding the embedding of input sentence obtained by the encoder module. $\hat{Y_i} = \{\hat{q_0}, \hat{q_1}, ..., \hat{q}_{T+1}\}$ are the predicted question tokens for the input X_i. Here, we are using $\hat{q_0}$ and $\hat{q}_{T+1}$ as the special START and STOP token respectively. The predicted question token ($\hat{q_i}$) is obtained by applying Softmax on the probability distribution $\hat{p_i}$. The question tokens at different time steps are given by the following equations where LSTM refers to the standard LSTM cell equations:

$$
\begin{aligned}
&d_{-1} = \text{Encoder}(f_i) \\
&h_0 = \text{LSTM}(d_{-1}) \\
&d_t = W_d * q_t, \forall t \in \{0, 1, 2, ...T-1\} \\
&h_{t+1} = \text{LSTM}(d_t, h_t), \forall t \in \{0, 1, 2, ...T-1\} \\
&\hat{p}_{t+1} = W_v * h_{t+1} \\
&\hat{q}_{t+1} = \text{Softmax}(\hat{p}_{t+1}) \\
&\text{Loss}_{t+1} = \text{loss}(\hat{q}_{t+1}, q_{t+1})
\end{aligned}
\tag{1}
$$

Where $\hat{q}_{t+1}$ is the predicted question token and q_{t+1} is the ground truth one. In order to capture local label information, we use the Cross Entropy loss which is given by the following equation:

$$
L_{local} = \frac{-1}{T} \sum_{t=1}^{T} q_t \text{logP}(\hat{q}_t|q_0, ..q_{t-1}) \tag{2}
$$

Here T is the total number of sentence tokens, $\text{P}(\hat{q}_t|q_0, ..q_{t-1})$ is the predicted probability of the sentence token, q_t is the ground truth token.

3.4 Discriminative-LSTM

The aim of the Discriminative-LSTM is to make the predicted sentence embedding f_i^p and ground truth sentence embedding f_i^g indistinguishable as shown in Figure 2. Here we pass $\hat{p_d}$ to the shared encoder-LSTM to obtain f_i^p and also the ground truth sentence to the shared encoder-LSTM to obtain f_i^g. The discriminator module estimates a loss function between the generated and ground truth paraphrases. Typically, the discriminator is a binary classifier loss, but here we use a global loss, similar to (Reed et al.,

2016) which acts on the last hidden state of the recurrent neural network (LSTM). The main objective of this loss is to bring the generated paraphrase embeddings closer to its ground truth paraphrase embeddings and farther from the other ground truth paraphrase embeddings (other sentences in the batch). Here our discriminator network ensures that the generated embedding can reproduce better paraphrases. We are using the idea of sharing discriminator parameters with encoder network, to enforce learning of embeddings that not only minimize the local loss (cross entropy), but also the global loss.

Suppose the predicted embeddings of a batch is $e_p = [f_1^p, f_2^p, ..f_N^p]^T$, where f_i^p is the sentence embedding of i^{th} sentence of the batch. Similarly ground truth batch embeddings are $e_g = [f_1^g, f_2^g, ..f_N^g]^T$, where N is the batch size, $f_i^p \in \mathbb{R}^d$ $f_i^g \in \mathbb{R}^d$. The objective of global loss is to maximize the similarity between predicted sentence f_i^p with the ground truth sentence f_i^g of i^{th} sentence and minimize the similarity between i^{th} predicted sentence, f_i^p, with j^{th} ground truth sentence, f_j^g, in the batch. The loss is defined as

$$L_{global} = \sum_{i=1}^{N}\sum_{j=1}^{N} \max(0, ((f_i^p \cdot f_j^g) - (f_i^p \cdot f_i^g) + 1)) \tag{3}$$

Gradient of this loss function is given by

$$\left(\frac{dL}{de_p}\right)_i = \sum_{j=1, j\neq i}^{N} (f_j^g - f_i^g) \tag{4}$$

$$\left(\frac{dL}{de_g}\right)_i = \sum_{j=1, j\neq i}^{N} (f_j^p - f_i^p) \tag{5}$$

3.5 Cost function

Our objective is to minimize the total loss, that is the sum of local loss and global loss over all training examples. The total loss is:

$$L_{total} = \frac{1}{M}\sum_{i=1}^{M}(L_{local} + L_{global}) \tag{6}$$

Where M is the total number of examples, L_{local} is the cross entropy loss, L_{global} is the global loss.

Dataset	Model	BLEU1	BLEU2	BLEU3	BLEU4	ROUGE	METEOR
50K	ED-L(Base Line)	33.7	22.3	18.0	12.1	35.3	14.3
	EDD-G	40.7	28.3	21.1	16.1	39.7	19.6
	EDD-LG	40.9	28.6	21.3	16.1	40.2	19.8
	EDD-LG(shared)	**41.1**	**29.0**	**21.5**	**16.5**	**40.6**	**20.1**
100K	ED-L(Base Line)	35.1	25.4	19.6	14.4	37.4	15.4
	EDD-G	42.1	29.4	21.6	16.4	41.4	20.4
	EDD-LG	44.2	31.6	22.1	17.9	43.6	22.1
	EDD-LG(shared)	**45.7**	**32.4**	**23.8**	**17.9**	**44.9**	**23.1**

Table 1: Analysis of variants of our proposed method on Quora Dataset as mentioned in section 4.1.3. Here L and G refer to the Local and Global loss and shared represents the parameter sharing between the discriminator and encoder module. As we can see that our proposed method EDD-LG(shared) clearly outperforms the other ablations on all metrics and detailed analysis is present in section 4.1.3.

4 Experiments

We perform experiments to better understand the behavior of our proposed embeddings. To achieve this, we benchmark Encoder Decoder Discriminator Local-Global (shared) (EDD-LG(shared)) embeddings on two text understanding problems, Paraphrase Generation and Sentiment Analysis. We use the Quora question pairs dataset [1] for paraphrase generation and Stanford Sentiment Treebank dataset (Socher et al.,

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

2013) for sentiment analysis. In this section we describe the different datasets, experimental setup and results of our experiments.

4.1 Paraphrase Generation

Paraphrase generation is an important problem in many NLP applications such as question answering, information retrieval, information extraction, and summarization. It involves generation of similar meaning sentences.

4.1.1 Dataset

We use the newly released Quora question pairs dataset for this task. It consists of over 400K potential question duplicate pairs. As pointed out in (Gupta et al., 2017), the question pairs having the binary value 1 are the ones which are actually the paraphrase of each other and the others are duplicate questions. So, we choose all such question pairs with binary value 1. There are a total of 149K such questions. Some examples of generated question-paraphrase pairs are provided in Table 3. More results are present in the appendix.

Dataset	Model	BLEU1	METEOR	TER
50K	Unsupervised VAE (Gupta et al., 2017)	8.3	12.2	83.7
	VAE-S (Gupta et al., 2017)	11.9	17.4	69.4
	VAE-SVG (Gupta et al., 2017)	17.1	21.3	63.1
	VAE-SVG-eq (Gupta et al., 2017)	17.4	**21.4**	61.9
	EDD-G (**Ours**)	40.7	19.7	51.2
	EDD-LG(**Ours**)	40.9	19.8	51.0
	EDD-LG(shared)(**Ours**)	**41.1**	20.1	**50.8**
100K	Unsupervised (Gupta et al., 2017)	10.6	14.3	79.9
	VAE-S (Gupta et al., 2017)	17.5	21.6	67.1
	VAE-SVG (Gupta et al., 2017)	22.5	24.6	55.7
	VAE-SVG-eq (Gupta et al., 2017)	22.9	**24.7**	55.0
	EDD-G (**Ours**)	42.1	20.4	49.9
	EDD-LG(**Ours**)	44.2	22.1	48.3
	EDD-LG(shared)(**Ours**)	**45.7**	23.1	**47.5**

Table 2: Analysis of Baselines and State-of-the-Art methods for paraphrase generation on Quora dataset. As we can see clearly that our model outperforms the state-of-the-art methods by a significant margin in terms of BLEU and TER scores. Detailed analysis is present in section 4.1.4. A lower TER score is better whereas for the other metrics, a higher score is better. Details for the metrics are present in the appendix.

4.1.2 Experimental Protocols

We follow the experimental protocols mentioned in (Gupta et al., 2017) for the Quora Question Pairs dataset. In our experiments, we divide the dataset into 2 parts 145K and 4K question pairs. We use these as our training and testing sets. We further divide the training set into 50K and 100K dataset sizes and use the rest 45K as our validation set. We also followed the dataset split mentioned in (Li et al., 2017b) to calculate the accuracies on a different test set and provide the results on our project webpage. We trained our model end-to-end using local loss (cross entropy loss) and global loss. We have used RMSPROP optimizer to update the model parameter and found these hyperparameter values to work best to train the Paraphrase Generation Network: learning rate $= 0.0008$, batch size $= 150, \alpha = 0.99, \epsilon = 1e-8$. We have used learning rate decay to decrease the learning rate on every epoch by a factor given by:

$$\text{Decay\_factor} = exp\left(\frac{log(0.1)}{a * b}\right)$$

where $a = 1500$ and $b = 1250$ are set empirically.

S.No	**Original Question**	**Ground Truth Paraphrase**	**Generated Paraphrase**
1	Is university really worth it?	Is college even worth it?	Is college really worth it?
2	Why India is against CPEC?	Why does India oppose CPEC?	Why India is against Pakistan?
3	How can I find investors for my tech startup?	How can I find investors for my startup on Quora?	How can I find investors for my startup business?
4	What is your view/opinion about surgical strike by the Indian Army?	What world nations think about the surgical strike on POK launch pads and what is the reaction of Pakistan?	What is your opinion about the surgical strike on Kashmir like?
5	What will be Hillary Clinton's strategy for India if she becomes US President?	What would be Hillary Clinton's foreign policy towards India if elected as the President of United States?	What will be Hillary Clinton's policy towards India if she becomes president?

Table 3: Examples of Paraphrase generation on Quora Dataset. We observe that our model is able to understand abbreviations as well and then ask questions on the basis of that as is the case in the second example.

4.1.3 Ablation Analysis

We experimented with different variations for our proposed method. We start with baseline model which we take as a simple encoder and decoder network with only the local loss (ED-Local) (Sutskever et al., 2014). Further we have experimented with encoder-decoder and a discriminator network with only global loss (EDD-Global) to distinguish the ground truth paraphrase with the predicted one. Another variation of our model is used both the global and local loss (EDD-LG). The discriminator is the same as our proposed method, only the weight sharing is absent in this case. Finally, we make the discriminator share weights with the encoder and train this network with both the losses (EDD-LG(shared)). The analyses are given in table 1. Among the ablations, the proposed EDD-LG(shared) method works way better than the other variants in terms of BLEU and METEOR metrics by achieving an improvement of 8% and 6% in the scores respectively over the baseline method for 50K dataset and an improvement of 10% and 7% in the scores respectively for 100K dataset.

4.1.4 Baseline and State-of-the-Art Method Analysis

There has been relatively less work on this dataset and the only work which we came across was that of (Gupta et al., 2017). We further compare our method EDD-LG(shared) model with their VAE-SVG-eq which is the current state-of-the-art on Quora datset. Also we provide comparisons with other methods proposed by them in table 2. As we can see from the table that we achieve a significant improvement of 24% in BLEU score and 11% in TER score (A lower TER score is better) for 50K dataset and similarly 22% in BLEU score and 7.5% in TER score for 100K dataset.

4.1.5 Statistical Significance Analysis

We have analysed statistical significance (Demšar, 2006) for our proposed embeddings against different ablations and the state-of-the-art methods for the paraphrase generation task. The Critical Difference (CD) for Nemenyi (Fišer et al., 2016) test depends upon the given α (confidence level, which is 0.05 in our case) for average ranks and N (number of tested datasets). If the difference in the rank of the two methods lies within CD, then they are not significantly different, otherwise they are statistically different. Figure 3 visualizes the post hoc analysis using the CD diagram. From the figure, it is clear that our embeddings work best and the results are significantly different from the state-of-the-art methods.

Model	Error Rate (Fine-Grained)
Naive Bayes (Socher et al., 2013)	59.0
SVMs (Socher et al., 2013)	59.3
Bigram Naive Bayes (Socher et al., 2013)	58.1
Word Vector Averaging (Socher et al., 2013)	67.3
Recursive Neural Network (Socher et al., 2013)	56.8
Matrix Vector-RNN (Socher et al., 2013)	55.6
Recursive Neural Tensor Network (Socher et al., 2013)	54.3
Paragraph Vector (Le and Mikolov, 2014)	51.3
EDD-LG(shared) (**Ours**)	**35.6**

Table 4: Performance of our method compared to other approaches on the Stanford Sentiment Treebank Dataset. The error rates of other methods are reported in (Le and Mikolov, 2014)

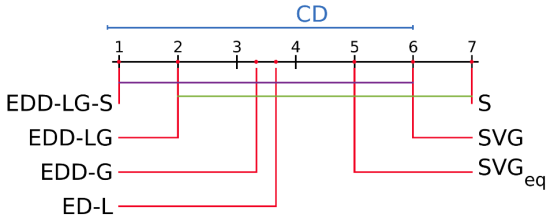

Figure 3: The mean rank of all the models on the basis of BLEU score are plotted on the x-axis. Here EDD-LG-S refers to our EDD-LG shared model and others are the different variations of our model described in section 4.1.3 and the models on the right are the different variations proposed in (Gupta et al., 2017). Also the colored lines between the two models represents that these models are not significantly different from each other. CD=5.199,p=0.0069

4.2 Sentiment Analysis with Stanford Sentiment Treebank (SST) Dataset

4.2.1 Dataset

This dataset consists of sentiment labels for different movie reviews and was first proposed by (Pang and Lee, 2005). (Socher et al., 2013) extended this by parsing the reviews to subphrases and then fine-graining the sentiment labels for all the phrases of movies reviews using Amazon Mechanical Turk. The labels are classified into 5 sentiment classes, namely {Very Negative, Negative, Neutral, Positive, Very Positive}. This dataset contains a total 126k phrases for training set, 30k phrases for validation set and 66k phrases for test set.

4.2.2 Tasks and Baselines

In (Socher et al., 2013), the authors propose two ways of benchmarking. We consider the 5-way fine-grained classification task where the labels are {Very Negative, Negative, Neutral, Positive, Very Positive}. The other axis of variation is in terms of whether we should label the entire sentence or all phrases in the sentence. In this work we only consider labeling all the phrases. (Socher et al., 2013) apply several methods to this dataset and we show their performance in table 4.

4.2.3 Experimental Protocols

For the task of Sentiment analysis, we are using a similar method of performing the experiments as used by (Socher et al., 2013). We treat every subphrase in the dataset as a separate sentence and learn their corresponding representations. We then feed these to a logistic regression to predict the movie ratings. During inference time, we used a method simialr to (Le and Mikolov, 2014) in which we freeze the representation of every word and use this to construct a representation for the test sentences which are then fed to a logistic regression for predicting the ratings. In order to train a sentiment classification model, we have used RMSPROP, to optimize the classification model parameter and we found these hyperparameter

Phrase ID	Phrase	Sentiment
162970	The heaviest, most joyless movie	Very Negative
159901	Even by dumb action-movie standards, Ballistic : Ecks vs. Sever is a dumb action movie.	
158280	Nonsensical, dull "cyber-horror" flick is a grim, hollow exercise in flat scares and bad acting	
159050	This one is pretty miserable, resorting to string-pulling rather than legitimate character development and intelligent plotting.	
157130	The most hopelessly monotonous film of the year, noteworthy only for the gimmick of being filmed as a single unbroken 87-minute take.	
156368	No good jokes, no good scenes, barely a moment	Negative
157880	Although it bangs a very cliched drum at times	
159269	They take a long time to get to its gasp-inducing ending.	
157144	Noteworthy only for the gimmick of being filmed as a single unbroken 87-minute	
156869	Done a great disservice by a lack of critical distance and a sad trust in liberal arts college bumper sticker platitudes	
221765	A hero can stumble sometimes.	Neutral
222069	Spiritual rebirth to bruising defeat	
218959	An examination of a society in transition	
221444	A country still dealing with its fascist past	
156757	Have to know about music to appreciate the film's easygoing blend of comedy and romance	
157663	A wildly funny prison caper.	Positive
157850	This is a movie that's got oodles of style and substance.	
157879	Although it bangs a very cliched drum at times, this crowd-pleaser's fresh dialogue, energetic music, and good-natured spunk are often infectious.	
156756	You don't have to know about music to appreciate the film's easygoing blend of comedy and romance.	
157382	Though of particular interest to students and enthusiast of international dance and world music, the film is designed to make viewers of all ages, cultural backgrounds and rhythmic ability want to get up and dance.	
162398	A comic gem with some serious sparkles.	Very Positive
156238	Delivers a performance of striking skill and depth	
157290	What Jackson has accomplished here is amazing on a technical level.	
160925	A historical epic with the courage of its convictions about both scope and detail.	
161048	This warm and gentle romantic comedy has enough interesting characters to fill several movies, and its ample charms should win over the most hard-hearted cynics.	

Table 5: Examples of Sentiment classification on test set of kaggle competition dataset.

values to be working best for our case: learning rate $= 0.00009$, batch size $= 200, \alpha = 0.9, \epsilon = 1e-8$.

4.2.4 Results

We report the error rates of different methods in table 4. We can clearly see that the performance of bag-of-words or bag-of-n-grams models (the first four models in the table) is not up to the mark and instead the advanced methods (such as Recursive Neural Network (Socher et al., 2013)) perform better on sentiment analysis task. Our method outperforms all these methods by an absolute margin of 15.7% which is a significant increase considering the rate of progress on this task. We have also uploaded our models to the online competition on Rotten Tomatoes dataset [2] and obtained an accuracy of 62.606% on their test-set of 66K phrases.

We provide 5 examples for each sentiment in table 5. We can see clearly that our proposed embeddings are able to get the complete meaning of smaller as well as larger sentences. For example, our model classifies 'Although it bangs a very cliched drum at times' as Negative and 'Although it bangs a very cliched drum at times, this crowd-pleaser's fresh dialogue, energetic music, and good-natured spunk are often infectious.' as positive showing that it is able to understand the finer details of language. More results and visualisations showing the part of the phrase to which the model attends while classifying are present in the appendix. The link for the project website and code is provided here [3].

[2] website: `www.kaggle.com/c/sentiment-analysis-on-movie-reviews`

[3] Project website: `https://badripatro.github.io/Question-Paraphrases/`

5 Conclusion

In this paper we have proposed a sentence embedding using a sequential encoder-decoder with a pairwise discriminator. We have experimented with this text embedding method for paraphrase generation and sentiment analysis. We also provided experimental analysis which justifies that a pairwise discriminator outperforms the previous state-of-art methods for NLP tasks. We also performed ablation analysis for our method, and our method outperforms all of them in terms of BLEU, METEOR and TER scores. We plan to generalize this to other text understanding tasks and also extend the same idea in vision domain.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. of ACL workshop on Intrinsic and Extrinsic Evaluation measures for Machine Translation and/or Summarization*, volume 29, pages 65–72.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. pages 1724–1734.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. 1:1342–1352.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2016. Janes v0. 4: Korpus slovenskih spletnih uporabniških vsebin. *Slovenščina*, 2(4):2.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

E Grefenstette, G Dinu, Y Zhang, M Sadrzadeh, and M Baroni. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 131–142.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2017. A deep generative framework for paraphrase generation. *arXiv preprint arXiv:1709.05074*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Sadid A Hasan, Bo Liu, Joey Liu, Ashequl Qadir, Kathy Lee, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. 2016. Neural clinical paraphrase generation with attention. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 42–53.

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in nlp. In *Proceedings of NAACL-HLT*, pages 681–691.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017b. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

Xiaodan Liang, Zhiting Hu, Hao Zhang, Chuang Gan, and Eric P Xing. 2017. Recurrent topic-transition gan for visual paragraph generation. *CoRR, abs/1703.07022*, 2.

Dekang Lin and Patrick Pantel. 2001. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Rui Lin, Shujie Liu, Muyun Yang, Mu Li, Ming Zhou, and Sheng Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 899–907.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out:Proceedings of the ACL-04 workshop*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190. Association for Computational Linguistics.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(4):694–707.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1577–1586.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation.

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)*, pages 129–136.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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

Peter D Turney and Michael L Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. corr abs/1609.08144.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrasing headlines by machine translation: Sentential paraphrase acquisition and generation using google news. *LOT Occasional Series*, 16:169–183.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 172–182. Association for Computational Linguistics.

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*.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2972–2978. AAAI Press.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1263–1271. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*.

A Appendix

A.1 Quantitative Evaluation

We use automatic evaluation metrics which are prevalent in machine translation domain: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE-n (Lin, 2004) and Translation Error Rate (TER) (Snover et al., 2006). These metrics perform well for Paraphrase generation task and also have a higher correlation with human judgments (Madnani et al., 2012; Wubben et al., 2010). BLEU uses n-gram precision between the ground truth and the predicted paraphrase. considers exact match between reference whereas ROUGE considers recall for the same. On the other hand, METEOR uses stemming and synonyms (using WordNet) and is based on the harmonic mean of unigram-precision and unigram-recall. TER is based on the number of edits (insertions, deletions, substitutions, shifts) required to convert the generated output into the ground truth paraphrases and quite obviously a lower TER score is better whereas other metrics prefer a higher score for showing improved performance. We provided our results using all these metrics and compared it with existing baselines.

A.2 Paraphrase Generation

Here we provide some more examples of the paraphrase generation task in table 6. Our model is also able to generate sentences which capture higher level semantics like in the last example of table 6.

S.No	**Original Question**	**Ground Truth Paraphrase**	**Generated Paraphrase**
1	How do I add content on Quora?	How do I add content under a title at Quora?	How do I add images on Quora ?
2	Is it possible to get a long distance ex back?	Long distance relationship: How to win my ex-gf back?	Is it possible to get a long distance relationship back ?
3	How many countries are there in the world? Thanks!	How many countries are there in total?	How many countries are there in the world ? What are they ?
4	What is the reason behind abrupt removal of Cyrus Mistry?	Why did the Tata Sons sacked Cyrus Mistry?	What is the reason behind firing of Cyrus Mistry ?
5	What are some extremely early signs of pregnancy?	What are the common first signs of pregnancy? How can I tell if I'm pregnant? What are the symptoms?	What are some early signs of pregnancy ?
6	How can I improve my critical reading skills?	What are some ways to improve critical reading and reading comprehension skills?	How can I improve my presence of mind ?

Table 6: Examples of Paraphrase generation on Quora Dataset.

A.3 Sentiment Analysis

We also provide visualization of different parts of the sentence on which our model focuses while predicting the sentiment in Figure 4 and some more examples of the Sentiment analysis task on SST dataset in Table 7.

A.3.1 Sentiment Visualization of the sentence

(Li et al., 2016a) have proposed a mechanism to visualize language features. We conducted a toy experiment for our EDD-LG(shared) model. Figure 4 represents saliency heat map for EDD-LG(shared) model sentiment analysis. We obtained 60 dimensional feature maps for each word present in the target sentence. The heat map captures the measure of influence of the sentimental decision. In the heat map, each word of a sentence (from top to bottom, first word at top) represents its contribution for making the sentimental decision. For example in the first image in 4, the word 'comic' contributed more (2nd word,

row 10-20). Similarly in the second image, first, second, and third (‘A’,‘wildly’,‘funny’) words have more influence for making this sentence have a positive sentiment.

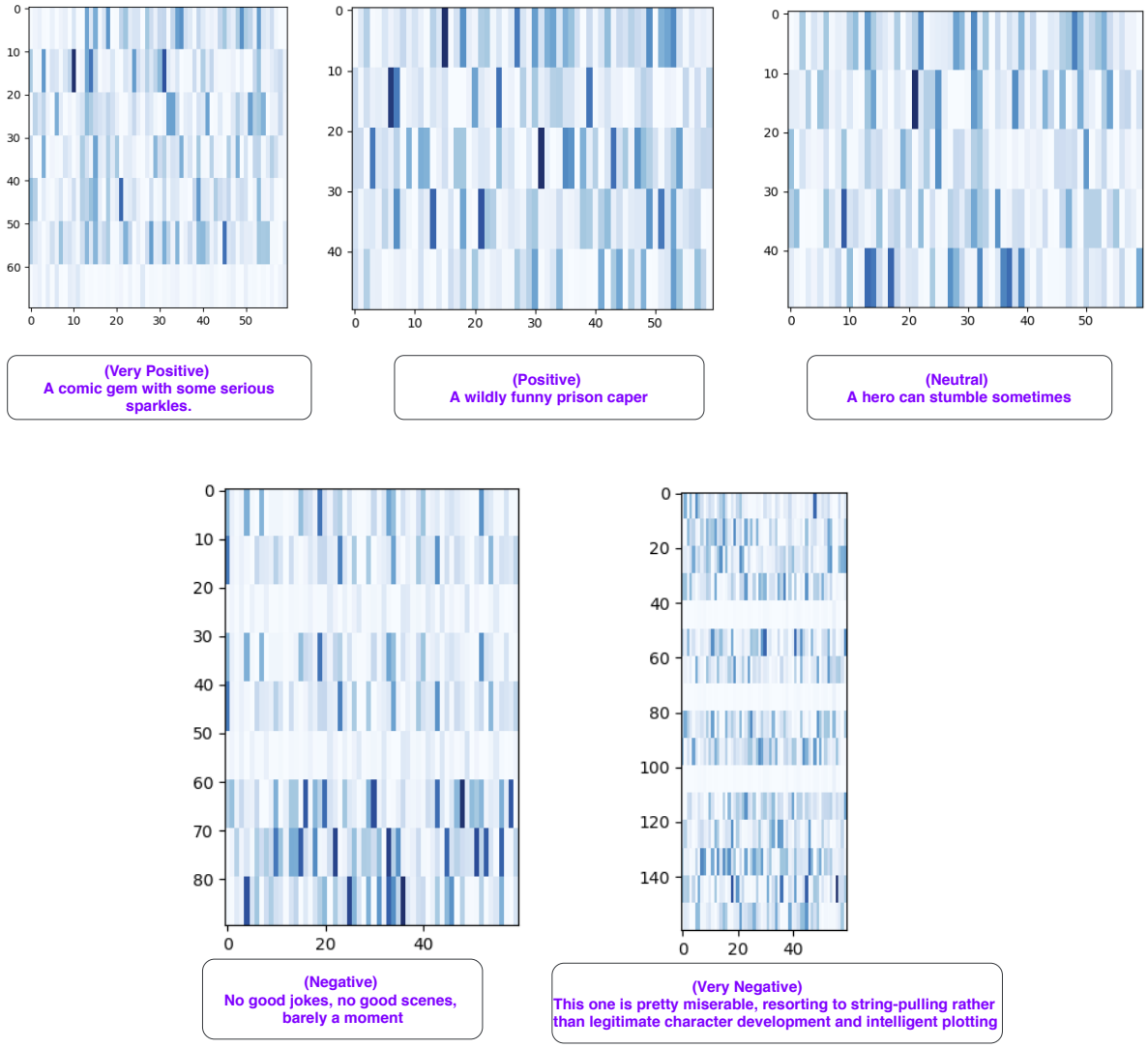

Figure 4: These are the visualisations for the sentiment analysis for some examples and we can clearly see that our model focuses on those words which we humans focus while deciding the sentiment for any sentence. In the second image, ‘wildly’ and ‘funny’ are emphasised more than the other words.

Phrase ID	Phrase	Sentiment
156628	The movie is just a plain old monster	Very Negative
157078	a really bad community theater production of West Side Story	
159749	Suffers from rambling , repetitive dialogue and the visual drabness endemic to digital video .	
163425	The picture , scored by a perversely cheerful Marcus Miller accordionharmonicabanjo abomination , is a monument to bad in all its florid variety .	
163483	lapses quite casually into the absurd	
163882	It all drags on so interminably it 's like watching a miserable relationship unfold in real time .	
164436	Your film becomes boring , and your dialogue is n't smart	
165179	Another big , dumb action movie in the vein of XXX , The Transporter is riddled with plot holes big enough for its titular hero to drive his sleek black BMW through .	
156567	It would be hard to think of a recent movie that has worked this hard to achieve this little fun	Negative
156689	A depressing confirmation	
157730	There 's not enough here to justify the almost two hours.	
157695	a snapshot of a dangerous political situation on the verge of coming to a head	
158814	It is ridiculous , of course	
159281	A mostly tired retread of several other mob tales.	
159632	We are left with a superficial snapshot that , however engaging , is insufficiently enlightening and inviting .	
159770	It 's as flat as an open can of pop left sitting in the sun .	
156890	liberal arts college bumper sticker platitudes	Neutral
160247	the movie 's power as a work of drama	
160754	Schweig , who carries the film on his broad , handsome shoulders	
160773	to hope for any chance of enjoying this film	
201255	also examining its significance for those who take part	
201371	those who like long books and movies	
221444	a country still dealing with its fascist past	
222102	used to come along for an integral part of the ride	
157441	the film is packed with information and impressions .	Positive
157879	Although it bangs a very cliched drum at times , this crowd-pleaser 's fresh dialogue , energetic music , and good-natured spunk are often infectious.	
157663	A wildly funny prison caper.	
157749	This is one for the ages.	
157806	George Clooney proves he 's quite a talented director and Sam Rockwell shows us he 's a world-class actor with Confessions of a Dangerous Mind .	
157850	this is a movie that 's got oodles of style and substance .	
157742	Kinnear gives a tremendous performance .	Very Positive
160562	The film is painfully authentic , and the performances of the young players are utterly convincing .	
160925	A historical epic with the courage of its convictions about both scope and detail.	
161048	This warm and gentle romantic comedy has enough interesting characters to fill several movies , and its ample charms should win over the most hard-hearted cynics .	
161459	is engrossing and moving in its own right	
162398	A comic gem with some serious sparkles .	
162779	a sophisticated , funny and good-natured treat , slight but a pleasure	
163228	Khouri then gets terrific performances from them all .	

Table 7: Examples of Sentiment classification on test set of kaggle dataset.