Transfusion: Efficient Information Extraction for Better Text Classification

Vaishali Singh
20800957
v47singh@uwaterloo.ca

***Abstract*—Text classification is the basis for many other natural language processing (NLP) tasks. Over the years, recurrent neural networks (RNNs) have emerged as one of the best models for NLP tasks. In this paper, we introduce a novel method: Transfusion, to extract information efficiently for better text classification. We've applied it to Bi-directional LSTM (Long short term memory), which is a variant of RNNs. The proposed model uses a transfusion layer and extracts meaningful information by combining the outputs of Bi-LSTM (with and without Attention) cell. We compare the performance of our method with other models, where, the output (both, forward and backward) is combined by either summing them, taking average, concatenating, or by applying a 1-D max-pooling. We evaluate our method on two datasets, namely, Standford Sentiment Treebank (SST1) and 20 News Group (20NG).**

***Index Terms*—BiLSTM, Text Classification, Regularization, Embeddings, Feature vectors, Information Extraction**

I. Introduction

Text is one of the most challenging modalities to deal with due to its unstructured nature. A lot of information is present in just a few set of words. This makes the task of developing good encoding mechanisms hard and time-consuming.

Text classification is a vital downstream NLP task. It has many applications such as, sentiment analysis, spam detection, text-to-speech conversion and few others. Text classification is a way of assigning the sentences to suitable predefined category or labels based on its content. Due to this, it has attracted many researchers and a variety of models have been proposed for the same.

One of the traditional methods introduced for encoding text was the bag-of-words(BoW) model. According to Wang and Manning et al., 2012, it treats texts as an unordered sets of words, however, it fails to encode word-order and syntactic features. Recently, deep learning models have achieved tremendous success for text classification and have shown more significant progress compared with BoW models because of the ability to deal with word order. Among the deep learning models, RNN have proved to be much suitable for this task as it can handle variable-length text such as sentences, phrases and documents. [1]

The RNN represent the given text in the form of token-vector matrix where each token is the id given to the word present in the given sentence. This vector matrix includes two dimensions: the time-step dimension and the feature vector dimension and in the process of learning feature representation, it will be updated. Now to obtain a fixed-length vector which is a weight representation over this matrix, that is time-step dimension as well as feature-vector dimension, many methods were introduced.

One of such methods was introduced by Lai et al., 2015 in which RNN utilizes 1D max pooling operation or attention-based operation introduced by Zhou et al., 2016, only over the time-step dimension of the matrix which extracts maximum values or generates a weighted representation to obtain a fixed-length vector. [2] [3]

Another model was introduced by Zhou, Peng, et al., 2016 which was Bidirectional Long Short-Term Memory Networks with TwoDimensional Max Pooling (BLSTM-2DPooling) to capture features on both the time-step dimension and the feature

vector dimension to obtain a fixed-length vector. This paper also applies 2D convolution (BLSTM-2DCNN) to capture more meaningful features to represent the input text. [4]

The process of text classification can either be improved by maximizing the meaningful information which will be passed onto the classification layer or by improving the classification method. In our paper, we will try to achieve the former target in such a way that we do not require heavily trained neural networks.

Therefore, this paper has introduced a light weight method which will combine the information in most efficient manner. Then we will compare our model with the already proposed models. The contributions of this paper can be summarized as follows:

- This paper proposes a combined framework, which utilizes BiLSTM-transfusion, to capture long-term sentence dependencies, and better extracts features with the help of transformation layer.

- We introduce a novel auxiliary loss to maximize mutual information between two distributions. Compared with the other models, BiLSTM-transfusion achieves better performance on both the datasets. In other words, it achieves better accuracy on Stanford Sentiment Treebank binary classification and fine-grained classification tasks.

- To better understand the effect of auxiliary loss function on the introduced transfusion layer, we have conducted experiments on Stanford Sentiment Treebank fine-grained task and 20Newsgroup dataset by using the model (Bi-LSTM with transfusion layer) with and without the loss function.

The rest of the paper is organized as follows: In Section 2, other methods used for doing text classification is reviewed. Section 3 presents the proposed model, its architecture and working in detail. Section 4 describes details about the implementation done for smoothly conducting the experiment. Section 5 presents the experimental results and their discussion. In section 6, the conclusion of the overal study is drawn and in the final section some improvements and future work is listed to do better analysis.

II. Literature Review

Text classification is one of the many application of NLP which have been widely researched since it has introduced. Many models have been introduced to improve text classification. For the classification purpose many machine learning based models such as Naive Bayes, Logistic regression and Support vector machines were used. But the accuracy of prediction depends on the features fed to these classifiers.

Since we cannot pass the text as it is to the machine learning model, it is first converted into numerical representations. Also the text can be of variable lengths that is long as well short. Wang and Manning, 2012 used bag-of-words model to obtain the text representation which treat text as an unordered set of words. This model lacks the semantics of the words. [1]

To overcome this limitation, Convolutional Neural Networks (CNN) were introduced by Kalchbrenner et al., 2014; Kim, 2014 utilizes 1D convolution to perform the feature mapping, and then applies 1D max pooling operation over the time-step dimension to obtain a fixed-length output. But to obtain the feature representations for variable length text such as sentences, short summaries and documents, recurrent neural networks were introduced. [5] [6]

Earlier the RNN used to obtain the fixed length vector after combining the two dimension by simply summing, averaging or concatenating the two vectors. Extracting features in a more meaningful way became another task to improve text classification. Then another model was introduced by Lai et al. 2015 which was RNN(recurrent neural network) along with 1D pooling was able to capture contextual information and thus resulted in a more meaningful feature

representation compared to others. [2]

Similar to the RNN-1D pooling, another model was introduced by Zhou et al., 2016, which was Attention-based RNN model. But both these models suffered from the drawback that they focused only on the time-step dimension of feature matrix. None of these models focus on the fact that feature vectors can also play important role in improving text classification. [3]

To overcome this limitation, Zhou, Peng, et al. proposed a model, that is, Bidirectional Long Short-Term Memory Networks with TwoDimensional Max Pooling (BiLSTM-2DPooling) to capture features on both the time-step dimension and the feature vector dimension. It first utilizes Bidirectional Long Short-Term Memory Networks (BLSTM) to transform the text into vectors and then 2D max pooling operation is done to obtain a fixed-length vector. This paper also applies 2D convolution (BiLSTM-2DCNN) to capture more meaningful features to represent the input text. [4]

Among the reviewed models, the BiLSTM-2DCNN model was the most effective scheme to model sentence as well as documents. All these models were applied on SST-1 (Stanford sentiment Treebank) and 20 Newsgroup dataset and the result were compared. Our proposed model is similar to this model except that we will use Attention based-BiLSTM which has a simple transfusion layer along with auxiliary loss function. This small change is introduced to replace the heavily trained 2DCNN with a light weight model.

III. Our Method

Many deep learning models have been introduced for the tasks of text classification. RNN (Recurrent Neural Network) have been widely used for many NLP(Natural language Processing) tasks. Among these models, RNN are more effective in dealing with sequential data. For our purpose, we will use Bi-directional LSTM (Long Short Term Memory). The framework for the same is described as below:

BiLSTM (Bidirectional LSTM):
RNN or recurrent neural networks suffered from the drawback of not being able to learn long-term dependencies. To overcome this drawback, LSTM or long short term memory networks were introduced. They were able to learn the information for longer periods of time but the information at future state cannot be reached by current state. Not long after this, Bidirectional LSTM networks were introduced by Schuster and Paliwal in 1997. The main feature of Bi-directional LSTM is that they contain the information from both ends of the sequence. Thus, the network has more input information to predict the ouput. [17]

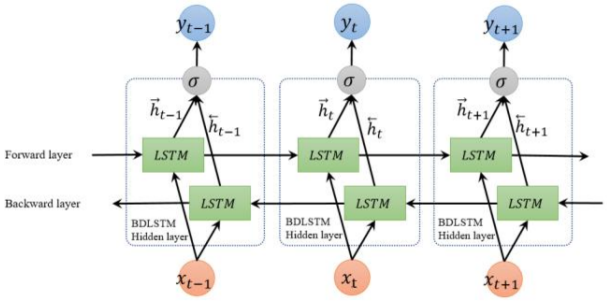

Fig. 1. A basic model explaining the architecture of Bi-Directional Long short temr memory(BiLSTM). [16]

Bi-directional LSTMs receives information from both the states, that is, past and future. This can be seen in the figure shown above. In other words, these networks(BiLSTM) split the neurons of a regular RNN into two directions, one for the positive time direction or forward states, and another for the negative time direction or backward states. Neither of these output states are connected to inputs of the opposite directions.

The input data from the past and future of the current time frame can be used to calculate the same output by employing two time directions simultaneously. This is the different from standard recurrent neural networks in which we need to provide an extra layer for including future information.Therefore, these networks are trained to simultaneously predict the positive and negative directions of time.

A single output is returned by the Bidirectional LSTM by connecting two hidden layers which are

running opposite to each other and due to this they are able to receive information from future as well as past states. In unsupervised or semi-supervised approaches, it is difficult to calculate a realistic probabilistic model. Therefore, his deep learning model or technique is more commonly used for supervised learning approaches.

The two directional neurons of Bi-LSTM do not interact with one another, therefore, the training algorithm for bi-directional LSTM is similar as LSTMs. The input and output layers cannot be updated at same time so we require some additional process, if we are doing the back-propagation.

According to the normal training algorithm, before passing the output neurons, we will first process both the forward and backward states in the "forward" pass. Then for doing the "backward" pass, first we process the output neurons and after this we will pass both the forward and backward states. Finally, we will update the weights once both these "forward" as well "backward" passes are complete. These networks are very useful when we need to extract the context of the data(text).

Attention Module: The main objective behind this mechanism is to automatically detect and focus on those words of text which will essentially help in predicting the target word. Attention is needed because if we have sentences that are longer then it will become hard for neural networks to squash all the information of a source sentence, regardless of its length, into a fixed-length vector. A significant advantage of this approach is that instead of translating the complete input sentence into a single fixed-length vector, it encodes the sentence into a sequence of vectors and while decoding the translation returns a subset of these vectors adaptively. [14]

Other models: Before we introduce our proposed model, we will describe the simple methods that were used with Attention based-BiLSTM to obtain a meaningful feature vector.

1) BiLSTM with SUM: In this model, Attention based BiLSTM is used to obtain Feature vector which will formed by adding or summing up all the outputs of the network.
2) BiLSTM with Average: In this model, Attention based BiLSTM is used to obtain Feature vector which will formed by taking the mean or average all the outputs of the network instead of simply adding them.
3) BiLSTM with Concatenation: In this model, Attention based BiLSTM is used to obtain Feature vector which will formed by concatenating all the outputs of the network.
4) BiLSTM with 1D pooling: Instead of using Attention, 1D pooling is applied on the time-step dimension returned by BiLSTM. The output of this 1D pooling layer is then used to obtain Feature vector.

Proposed Method:
The bidirectional LSTM model described above has been used till now. In this paper, we have proposed a new way to efficiently extract information in order to improve classification performance. For doing so, we introduce a transfusion layer along with an auxiliary loss function before the classification layer. The architecture of our proposed model can be seen in the figure 2.

Instead of defining a deterministic function to model the dynamics between the forward and backward direction, we will introduce a transfusion layer that will help the network decide for itself how to model the inter-directional dynamics. And in order to make sure that this transformation indeed contains information from both forward and backward direction we will introduce an auxiliary loss which will maximize the transformed vector and the concatenated vector. The two main components of the proposed model are discussed below in detail.

1) *Transfusion Layer:* The work of this transfusion layer is to combine the information from the both the directions ('2*h'), that is, forward dimension and backward dimension. The transfusion layer is a single layer perceptron in our case and is applied while fusing both, the encoder

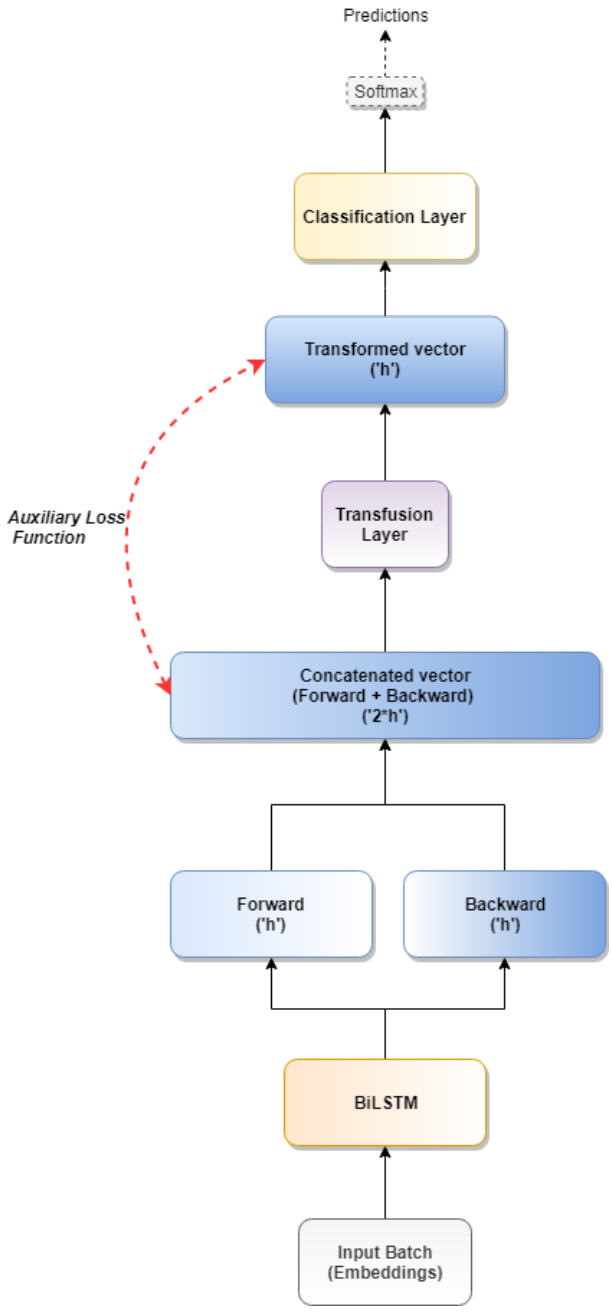

Fig. 2. Architecture of the proposed model.

outputs as well as the hidden vectors. This layer then provides a output which is of the dimension ('h'). The output vector is information enriched from both the forward and backward directions.

2) *Auxiliary Loss:* To maximize the mutual information between the output vector and the concatenated vector which was the input of transfusion layer, we will calculate the loss using the auxiliary loss function. Here, the auxiliary loss function is a mean square error. We have done zero padding in the output vector while calculating the loss to match the dimension of output vector with the concatenated one. This zero padding will additionally make sure that the extracted information is not biased towards the either of the directions. That is, it contains the information from both the vectors instead of trivially outputting either of these vectors. The auxiliary loss in our case is given by:

$$\mathcal{L}_{aux} = MSE([h_{tf}; \mathbf{0}], [h_{fw}; h_{bw}]), \qquad (1)$$

where $h_{tf} \in \mathcal{H}^{b \cdot h_d}$ represents the transfused vector, $\mathbf{0}$ is a null vector, $h_{fw}, h_{bw} \in \mathcal{H}^{b \cdot h_d}$ represent the forward and backward direction outputs of the LSTM vectors, and $[;]$ represents concatenation. Furthermore, b denotes batch size and h_d represents the number of hidden units.

Working of the Model: In the proposed model, we first pass the processed data through the Bi-LSTM layer. The Bi-LSTM layer will return a time-step dimension and hidden state vector each of size '2h' as output. Then the two elements of the output,that is forward and backward, are concatenated and we obtain concatenated vector(C') of size '2*h'. This vector 'C' is passed through the transfusion layer introduced in the model. Now, this layer will try to learn the information present in the concatenated vector and results in an output which is named as transformed vector of size 'h'.

Before calculating the auxiliary loss, zero padding is done at transformed layer as discussed above. Now this auxiliary loss function introduced along with the transfusion layer will try to maximize the information between transformed vector and forward and backward time-steps. So, the learning of layer is not biased towards any of the forward and backward vectors.

The transformed vector is the final representation of the input text. After obtaining the transformed vector, it will be passed onto the classification layer for doing the predictions. The classification layer is

further mapped with another softmax layer which calculates the estimated probability for each label. Finally, the label with highest probability is assigned to the text.

IV. Implementation

In this section, we will describe the dataset used, its pre-processing, training of model and the hyper-parameters used for implementing the proposed model to obtain the results which we have discussed in the later section.

Dataset: The proposed model has been tested mainly two datasets:

- *SST-1:* Stanford Sentiment Treebank or SST consists of fine grained sentiment labels for 215,154 phrases or sentences. The dataset contains five labels which are given as very negative, negative, neutral, positive and very positive. The corpus is based on the dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. It was parsed with the Stanford parser (Klein and Manning, 2003) and includes a total of 215,154 unique phrases from those parse trees. [8] [9]

- *20Newsgroup:* The 20Ng dataset contains messages from twenty newsgroups. We use the bydate version pre-processed by Cachopo (2007). For our model we have selected four major categories only (comp, politics, rec and religion) followed by Hingmire et al. (2013). [11] [10]

Data Pre-processing: Data pre-processing is done to convert the raw data into a processed one which will be fed to the model. For processing, we have divided the data into train and test dataset. Each of them contains the sentences along with their labels assigned.

Word Embeddings: The word embeddings are pre-trained on much larger corpus to achieve better generalization given limited amount of training data (Turian et al., 2010). In our proposed models we have utilized the word2vec embeddings created by a team of researchers led by Tomas Mikolov at Google in 2013. The words that are not present in the set of pre-trained words, are initialized by random sampling from uniform distribution. Apart from this, to improve the performance of classification we have done some fine-tuning on the word embeddings during training. [15]

Training the model: The model is then trained on the processed data and once the training is completed we can use it for doing prediction of the testing data. We break the given sentence into words. Then these words are mapped with token ids. Using these token ids, we obtain the vectors from the word embeddings. Then the matrix so obtained is called embedding matrix. The model here is Bi-directional LSTM which is implemented using Pytorch and this model was trained on Nvidia 1080Ti 12 GB RAM.

Hyper-parameters Tuning: For evaluating the model on SST1 and 20Newsgroup datasets we have used accuracy, F1-score, Recall and Precision metrica. The final hyper parameters that have been set for conducting this experiment as described below.

The dimension of word embeddings and the hidden units of LSTM are set as 300. Maximum sentence length and batch size is 30 and 64 respectively. The optimizer used for this experiment is AdaDelta with the default learning rate of 1.0 and weight decay of e^{-5}. And finally cross-entropy loss function with its reduction parameters set to 'sum'. For regularization, we have used Dropout operation (Hinton et al., 2012) with dropout rate of 0.5 for the word embeddings and 0.5 for the BLSTM layer. [12]

These hyper-parameters have been set according to the ones specified in *Zhou Peng et al.*, 2016. For their experiments they calculated using grid search. The hyper-parameters can be further fine-tuned in order to achieve better performance. [4]

V. Result

Now we will analyse and discuss the performance of the proposed model on two datasets described in the above section. Then we will also compare

Model	SST1 (%)				20NG (%)			
	P	R	F1	A	P	R	F1	A
BiLSTM(-attn)	20.52 (17.79)	20.86 (18.48)	19.33 (18.23)	38.33 (37.37)	96.13 (88.11)	95.20 (87.81)	95.39 (87.74)	96.34 (88.85)
+ sum	20.64	20.89	20.45	40.05	95.98	95.47	94.61	96.13
+ avg	19.92	19.45	18.17	38.61	96.01	95.23	95.71	96.18
+ tf(ours)	20.12	19.34	20.07	40.16	96.57	96.53	96.13	96.47
+ tf+L1(ours)	21.38	22.52	19.89	45.38	94.58	94.88	95.48	94.96
+ Pool1D-attn	21.92	22.22	21.65	37.51	87.77	88.18	87.88	88.65

TABLE I

CLASSIFICATION RESULTS IN TERMS OF EVALUATION METRICS FOR THE DIFFERENT MODELS IMPLEMENTED IN THIS PAPER.

its performance with the other models described in the method section.

For comparing these models we have used the following evaluation metrics namely: accuracy, precision, F-score and recall. These metrics are calculated with the help of confusion matrix. All these evaluation metrics are described below.

Confusion Matrix: It is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known in advance. the four main terms that define this table are:

- True Positives (TP) - These are the correctly predicted positive values which means that the value of both actual class and predicted class is yes.
- True Negatives (TN) - These are the correctly predicted negative values which means that the value of both actual class is no and predicted class is no.
- False Positives (FP) When we predict the class to be true whereas the original or actual class is false.
- False Negatives (FN) When we predict the class to be false wheras the actual class is true.

Both False positives and false negatives occur when the actual class contradicts the predicted class.

1) *Accuracy:* Accuracy is defined as the fraction of number of predictions that our model got right compared to the total actual predictions. It can be defined formally as:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

2) *Precision:* Precision, also known as positive predictive value, is the proportion of positive results that truly are positive. Formally it can be defined as:

$$Precision = \frac{TP}{TP+FP}$$

3) *Recall:* Recall, also known as sensitivity, is the ability of a test to correctly identify positive results to get the true positive rate.

$$Recall = \frac{TN}{TP+FN}$$

4) *F-score:* The F score, also known as F1 score or F measure, is a measure of a tests accuracy. It is formally defined as:

$$Fscore = 2 * \frac{Precision*Recall}{Precision+Recall}$$

The table 1 lists all the evaluation metrics, that is, precision(P), recall(R), F1-score(F) and accuracy(A) for all the models used in this study on both datasets. The models listed in the table are: 1. BiLSTM(Bi-directional Long short term memory networks) with attention module(BiLSTM w/ attn), 2. BiLSTM-Attn with sum, 3. BiLSTM-Attn with average, 4. BiLSTM-Attn with transfusion layer, 5. BiLSTM-Attn with transfusion layer and loss function, 6. BiLSTM with One-dimensional Max Pooling without the attention module(BiLSTM 1D pooling w/o attn).

As we can observe from the table, our proposed

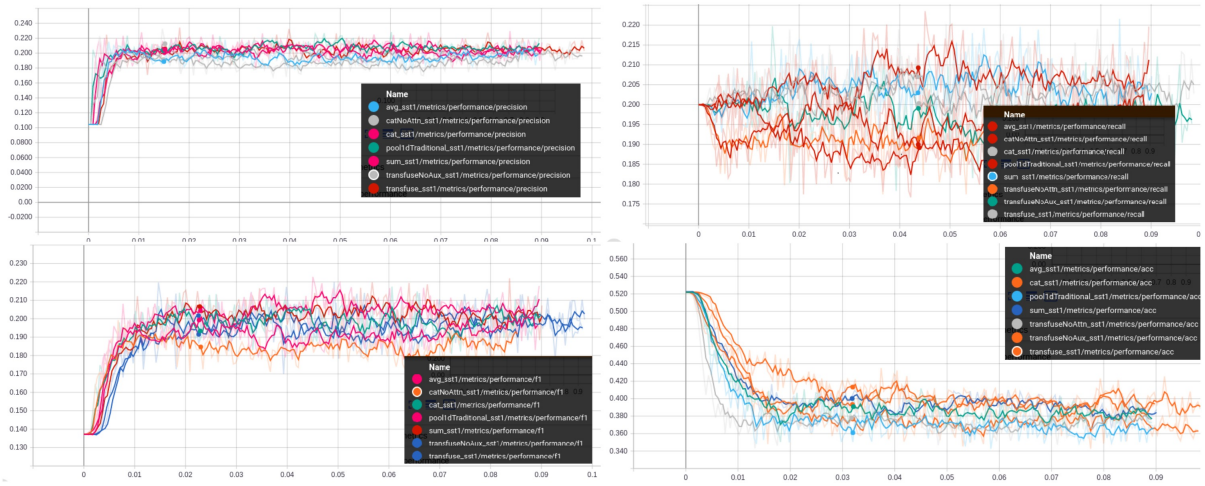

Fig. 3. Plotting the accuracy, precision, recall and f1-score for different models applied on SST1 dataset

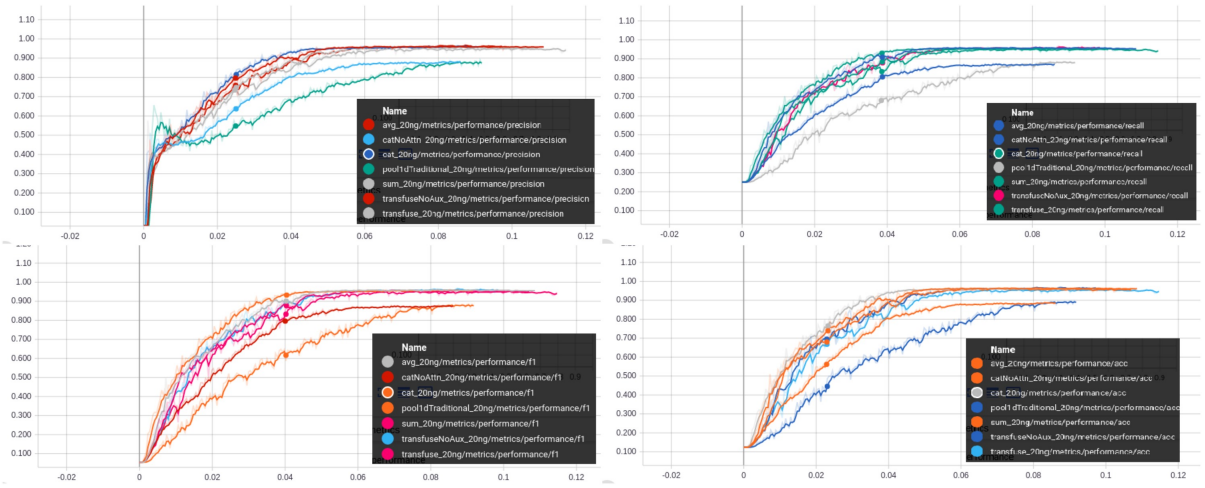

Fig. 4. Plotting the accuracy, precision, recall and f1-score for different models applied on 20Newsgroup dataset

model, that is, BiLSTM-Attn with transfusion and loss function has outperformed the other models in terms of accuracy (45.38%), precision (21.38%) and recall (22.52%) for dataset SST1. But for the other dataset, that is, 20 NewsGroup we observe a slight decrease. The reason can be that the dataset is not that complex. So introducing an additional loss function causes the model to overfit.

Effect of Auxiliary loss function on Transfusion For deciding the better model, we performed some ablation test to understand the affect of introducing a loss function over transfusion. The experiment was conducted on the SST1 dataset as this dataset is used for fine grained classification and shows reliable results. This can be seen from the result described above. We have also plotted a graph showing this which is shown in the figure below.

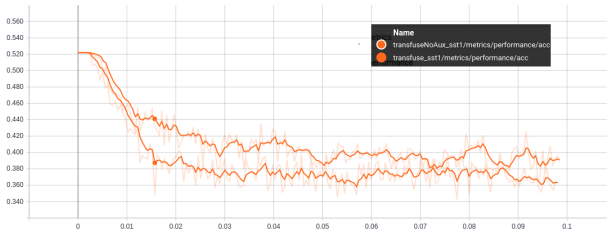

Fig. 5. Performance of Transfusion with and without auxiliary loss function on SST1 dataset.

We can evidently observe that auxiliary function has improved the learning of transfusion. Hence, it

is better to introduce this function to maximize the efficient extraction of information.

Figure 3 and Figure 4 shows the plot of precision(top-left), recall(top-right), f1-score(bottom-left) and accuracy(bottom-right) results of all the models for both datasets(SST1 and 20Ng). The trends of graphs in figure 3 clearly show that the transfusion model outperforms the others in case of SST1 dataset. Whereas for 20Ng dataset, we see a slight decrease in the performance of our model. This can be because the complexity of dataset is too low and it does not require any additional loss function. So if we see the trend of transfusion with loss function, this model starts to overfit and hence the performance of the model without loss function is better.

VI. Conclusion

We proposed a model for maximal efficient information extraction for better text classification with minimum effort. The method utilizes a transfusion layer with an auxiliary loss function. The loss function ensures to maximize the information between the transformed vector and forward and backward vectors. For comparing the performance of all the models we have used accuracy, precision, recall and f1-score as evaluation metrics. It was observed that our model outperformed on the SST1 dataset with highest accuracy, that is, 45.38%.

But for the other dataset, our model without the auxiliary loss function performed better than the one with loss function with the accuracy of 96.47% better than all other models. One of the possible reason can be because the 20Newsgroup dataset was simple and there was no requirement of applying any sort of regularization since the base model also had better accuracy. This case gives us an indication when there is no need of applying any kind of loss function or regularization. Because if we use any kind of auxiliary function than the model starts to overfit. The performance of our model can further be improved by fine-tuning the hyperparameters.

VII. Future Work

In this paper, we have tested our model on two datasets only, in future we would like to run the experiment on more robust datasets such as Yelp, or Amazon reviews. The proposed model is implemented only for text classification but it can be used for generation tasks also. Apart from this, other loss functions can be used in place of the introduced auxiliary loss function such as the mutual information measure. The hyperparameters can be fine-tuned to improve the performance of model used. In the future we want to compare our model with BiLSTM-2DCNN, so that we can see if the bulky neural network can be replaced with our light weight network. This will not only reduce the computation cost but will also bypass the distillation step we'd need to perform otherwise, thereby, saving us the effort of pre-training the model.

References

[1] Wang and Manning 2012, Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, pages 9094. Association for Computational Linguistics.

[2] Lei et al.2015, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding cnns for text: non-linear, nonconsecutive convolutions. arXiv preprint arXiv:1508.04112.

[3] Zhou et al.2016, Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In The 54th Annual Meeting of the Association for Computational Linguistics, page 207.

[4] Zhou, Peng, et al. "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling." arXiv preprint arXiv:1611.06639 (2016).

[5] Kalchbrenner et al.2014, Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[6] Kim2014, Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

[7] Dataset: Stanford Sentiment Treebank, https://nlp.stanford.edu/sentiment/index.html

[8] Pang and Lee2005, 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 115124. Association for Computational Linguistics.

[9] Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1 (ACL '03), Vol. 1. Association for Computational Linguistics, Stroudsburg, PA, USA, 423-430. DOI: https://doi.org/10.3115/1075096.1075150

[10] Ana Margarida de Jesus Cardoso Cachopo. 2007. Improving methods for single-label text categorization. Ph.D. thesis, Universidade Tecnica de Lisboa.

[11] Swapnil Hingmire, Sandeep Chougule, Girish K Palshikar, and Sutanu Chakraborti. 2013. Document classification by topic labeling. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pages 877880. ACM.

[12] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

[13] Confusion Matrix, https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

[14] Bahdanau et al., Neural machine translation by jointly learning to align and translate. ICLR, 2015.

[15] Turian, Joseph, Lev Ratinov, and Yoshua Bengio. "Word representations: a simple and general method for semi-supervised learning." Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, 2010.

[16] Image was obtained from this link,http://www.gabormelli.com/RKB/Bidirectional_Model

[17] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing 45.11 (1997): 2673-2681.

[18] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in neural information processing systems. 2015.