IITK at SemEval-2020 Task 10: Transformers for Emphasis Selection

Vipul Singhal* **Sahil Dhull*** **Rishabh Agarwal*** **Ashutosh Modi**
Indian Institute of Technology Kanpur (IITK)
{vipulsn,sahild,rish}@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

This paper describes the system proposed for addressing the research problem posed in Task 10 of SemEval-2020[1]: Emphasis Selection For Written Text in Visual Media. We propose an end-to-end model that takes as input the text and corresponding to each word gives the probability of the word to be emphasized. Our results show that transformer-based models are particularly effective in this task. We achieved the best Match$_m$ score (described in section 2.2) of 0.810 and were ranked third on the leader-board.

1 Introduction

Visual communication relies on images and short texts, e.g., in flyers, posters, etc. The purpose of these is to convey the message effectively and without ambiguity. Moreover, they should be able to attract the reader's attention within the first few seconds. For text, this can be achieved by laying emphasis on particular words to convey the intent better. The emphasis selection task is about designing automatic methods for choosing candidate words to be emphasized in short written texts, to enable automated design assistance in authoring (Figure 1). The dataset provided for this task is in English, and task description paper (Shirani et al., 2020) provided by the organizers describes the task, data, evaluation, results, and a summary of participating systems.

We tried two different approaches for the task. Our BiLSTM + Attention approach is inspired by the baseline paper (Shirani et al., 2019). In this approach, we tweaked the BiLSTM layers (Hochreiter and Schmidhuber, 1997), tried other layers like GRU (Cho et al., 2014), and used character embeddings along with word embeddings (Lample et al., 2016).

Our Transformers approach involves transfer learning using Transformer based models (Vaswani et al., 2017). This approach involves two types of models, the first one being a transformer-based model with the BiLSTM layer, the attention layer (Bahdanau et al., 2014), and fully connected layers on top. The second type of model involves transformer-based models with fully connected layers. We used BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), and GPT-2 (Radford et al., 2019) as transformer-based models. In the end, we tried the homogeneous and heterogeneous ensemble of these models.

Due to the nature of the problem, the systems which incorporate the whole context of a sentence would perform better. Further, the small size of the dataset was a bottleneck that can be countered by using transfer learning via the pre-trained models. Using transformer-based models accounts for both of these observations.

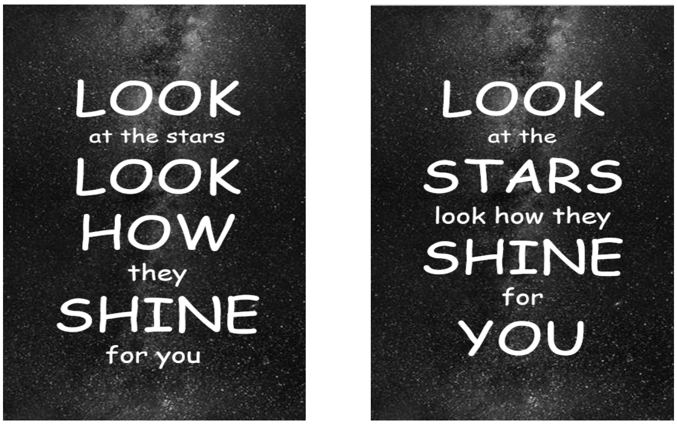
Our best submission achieved Match$_m$ score of 0.810 (Match$_m$ evaluation metric is described in Section 2.2) and was ranked third on the leaderboard. Our submissions under-performed in Score 1 (Match$_1$ as defined in Section 2.2) as compared to the other top-performing teams. Our code is available on Github[2].

* The authors equally contributed to this work.

[1] https://competitions.codalab.org/competitions/20815

[2] github.com/SahilDhull/emphasis_selection

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1665–1670
Barcelona, Spain (Online), December 12, 2020.



(a) Less Impact

(b) More Impact

Figure 1: Emphasizing different sets of words

2 Background

2.1 Problem definition

Given a sequence of words or tokens C = $\{x_1, x_2, ..., x_n\}$ in a text, the task is to compute a probabilistic score S_i for each x_i which indicates the degree of emphasis to be laid on the word.

2.2 Evaluation Metric

The evaluation metric for our problem is defined as follows:

For a given m (from 1 to 4), we first define 2 sets, $S_m^{(x)}$ - set of m words with top m probabilities according to ground truth and $\hat{S}_m^{(x)}$ - set of m words with top m probabilities according to model predictions. To get $S_m^{(x)}$, each word in the sentence has been manually annotated by 9 annotators using Amazon Mechanical Turk. More details regarding the same can be found in the baseline paper (Shirani et al., 2019).

Based on these 2 sets, we define Match$_m$ as

$$Match_m = \frac{\sum_{x \in D_{test}} \mid S_m^{(x)} \cap \hat{S}_m^{(x)} \mid /min(m, \mid x \mid)}{\mid D_{test} \mid}$$

where D_{test} is the dataset and x is text instance. We find Match$_m$ for m $\in \{1, .., 4\}$ and finally averaged to obtain the final score.

2.3 Data

We used the officially released dataset[3], which is the combination of the following two datasets:

Spark dataset: This dataset is a collection of short texts containing a variety of subjects featured in advertisements, posters, flyers, or motivational memes collected from Adobe Spark and contains 1200 instances.

Quotes dataset: This dataset collected from Wisdom Quotes contains 2,718 instances of quotes from well-known authors.

The dataset contains very short texts, usually fewer than 10 words, and is randomly divided into training (70%), development (10%) and test (20%) sets by the organizers.

2.4 Previous Work

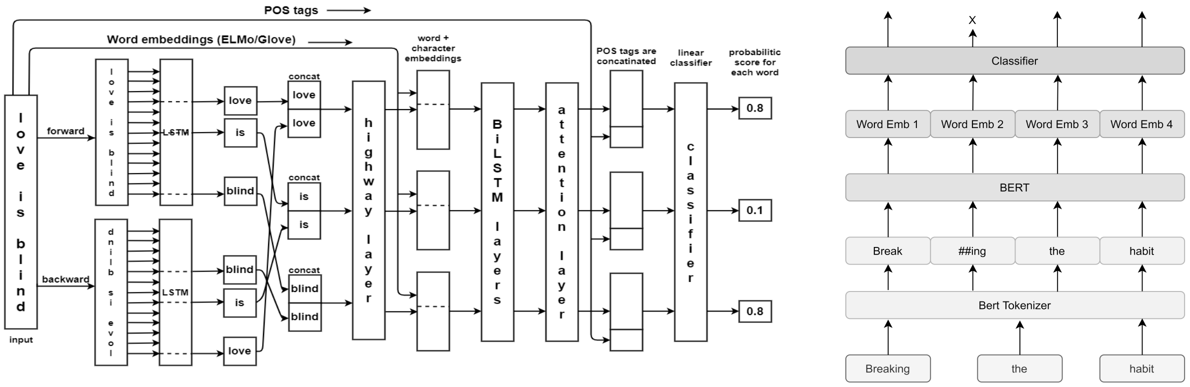
The baseline paper (Shirani et al., 2019) employs an end-to-end label distribution learning (LDL) and predicts a selection distribution. The model consists of an embedding layer, which is GloVe (Pennington

[3] https://github.com/RiTUAL-UH/SemEval2020 _Task10_Emphasis_Selection

et al., 2014) or ELMo (Peters et al., 2018) embeddings, followed by BiLSTM layers and, in the end, fully connected layers. Here, the output of BiLSTM layers for each word is passed through fully connected layers to predict the probabilities of emphasis and non-emphasis whose sum is 1.

3 System Overview

Our system takes as input the words in the text and corresponding to each word, gives the probability of the word to be emphasized. We tried two different types of sequence labeling approaches to learn emphasis patterns.



(a) BiLSTM + Attention Approach Model

(b) Transformers Approach Model

Figure 2: Models for both approaches

3.1 BiLSTM + Attention Approach

This approach involves character-level embeddings of each word of a sentence in addition to the word embeddings (Figure 2a). The characters of the sentences are passed through a pair of forward and backward LSTM. For each word, the outputs of the forward and backward LSTMs at the position of the last character of the word are taken, concatenated, and then passed through a highway layer to obtain the character-level embeddings of that word. These character-level embeddings are then concatenated with word embeddings obtained using pre-trained models such as GloVe or ELMo and passed through a pair of BiLSTM Layers. A self-attention layer (Cheng et al., 2016) which helps in learning the dependencies between the words in the sentence and gives different importance to different words while predicting is finally added, followed by a neural-network-based classifier which gives the probability of emphasis for each word. We also concatenated the POS tag of the words to the output of the Attention layers. The representation for each word at the output of Attention layers (along with POS tag concatenation) is passed through the fully connected layers to output the emphasis probability of the word.

3.2 Transformers Approach

In this approach, we use Transformer-based models with fully connected layers. Words are tokenized using appropriate tokenizer for each transformer model. Word embedding is obtained by concatenating the hidden layers of all encoder layers of the transformer-based model. And in the end, there are few fully connected layers with a dropout layer after each fully connected layer except the last one. Finally, the output of the last layer is passed through a sigmoid layer, which gives the probabilistic score of each word (Figure 2b). Here transformer-based models include BERT and BERT Large, RoBERTa and RoBERTa Large, XLNet Large, GPT-2 Medium, ALBERT, etc. We call this type of model as Transfomer-model and Classifier model (like BERT and Classifier model in Figure 2b) in all further discussions.

Finally, we used an ensemble of the above transformer models by taking the average of the scores predicted by these models. This helps to combine multiple models into one predictive model with less variance and improved predictions.

4 Experimental Setup

Our implementation uses PyTorch[4] library for deep learning models and the Transformers library by Hugging face[5] for the pre-trained transformer models and their tokenizers.

In BiLSTM + Attention approach, for obtaining character-level embeddings, a pair of BiLSTM layers with hidden size of 300 is used, whose output is passed through a highway layer and then concatenated with the GloVe or ELMo embeddings. This concatenated vector is then passed through another pair of BiLSTM layers with hidden dimension of 512. For the classifier, a pair of fully connected layer is used with ReLU activation function (Agarap, 2018) and hidden dimension of 20. Finally, the output of the last fully connected layer is passed through a sigmoid layer, which gives the probabilistic score of each word. To avoid overfitting, dropout layers with a probability of 0.3 were used.

In the Transformers approach, the pre-trained transformer models were used without freezing the layers, and the outputs of all the layers were concatenated. For the classifier, three fully connected layers are used with ReLU activation function and hidden dimension of 900 and 40 for larger transformer models and 300 and 20 for normal models. Dropout layers with a probability of 0.3 are also added to avoid overfitting.

In both approaches, Binary Cross-Entropy loss is used for training the model, whereas Match$_m$ score is used as the performance metrics for validation (as described in Section 2.2). We used Adam optimizer (Kingma and Ba, 2014) with the learning rate set to 0.001 for BiLSTM + Attention approach and 2e-5 for the Transformers approach. The model is fine-tuned for 100 epochs in BiLSTM + Attention approach and 30 epochs in the Transformers approach, and the reported test result corresponds to the best score obtained on the validation set.

5 Results

We attempted numerous small changes to our models in the BiLSTM + Attention as well as the Transformers approach to enhance the performance. This included hyper-parameter tuning like changing size and dimensions for different layers with some specific attempts particular to each approach.

Model	Best Score
Baseline Model	0.731
Baseline + Character Embedding Model	0.743
Baseline + Character Embedding Model + POS Tag Concatenation	0.747
Baseline + Character Embedding Model + POS Tag Concatenation + Language Model	0.738

Table 1: BiLSTM + Attention Approach

Model	Best Score
BERT + BiLSTM + Attention + FC layers	0.771
BERT + GRU + Attention + FC layers	0.755
BERT + Classifier	0.775
BERT_Large + Classifier	0.789
RoBERTa_Base + Classifier	0.775
RoBERTa_Large + Classifier	0.790
XLNet_Large + Classifier	**0.804**
ALBERT + Classifier	0.755
GPT-2 + Classifier	0.725
XLM-RoBERTa + Classifier	0.785

Table 2: Transformers Approach

For the BiLSTM + Attention approach, we tried training the model separately on Quotes and Spark dataset. We experimented using highway layers along with BiLSTM layers to attain a better outcome. We also tried using GRU layers (Yang et al., 2016) instead of BiLSTM layers. The word embeddings used in this approach are GloVe and ELMo embeddings. We also tried incorporating a language model to the character embedding model as done in this paper (Liu et al., 2018), where they predict the next word using forward and backward character RNN layers whenever they encounter space in the sentence. In Table 1, we present the best results (evaluation score as defined in section 2.2) on the validation set obtained after all these attempts for the BiLSTM + Attention approach. These were achieved using ELMo embeddings and BiLSTM layers.

[4] https://pytorch.org/

[5] https://huggingface.co/transformers/

For the Transformers approach, we altered the number of freezed layers in all the transformer models (BERT, XLNet, RoBERTa) while training. We attempted taking first-word embeddings/average of embeddings of all words in cases in which a word is broken into multiple tokens by the transformer model tokenizer. We also tried taking the output of the last hidden layer as well as concatenating all hidden layers of the transformer models as the final embedding of the word. Another type of model tried was the BERT Language Model with BiLSTM layers, attention, and fully connected layers on the top. This is basically using all the BERT hidden layers as embeddings by concatenating them, rather than GloVe or ELMo embeddings.

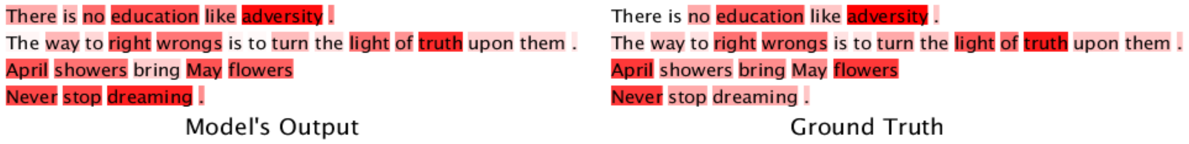
Best results were obtained by concatenating all layers, taking first token output as word embedding, without freezing any layer of the transformer model, and taking three fully connected layers on the top (their dimensions fine-tuned for each model). Table 2 contains the best results for all models on the validation set in the Transformers approach according to the evaluation metric.

Ensembling Model (runs)	**Validation Score**	**Test score**
XLNet (11)	**0.811**	0.807
BERT (2), RoBERTa (3), XLNet (8)	0.808	0.809
BERT (2), RoBERTa (3), XLNet (9)	0.810	**0.810**

Table 3: Ensembling Results

In the end, we tried an ensemble of models from the Transformers approach. We picked BERT_Large, RoBERTa_Large, and XLNet_Large models for both homogeneous and heterogeneous ensembling. After running multiple runs of each model, we took an average of the scores across multiple runs to obtain the final score for each word in each instance. Some of the results on the validation and test set are given in Table 3 (Large variants of all transformer models were used). The numbers in the bracket denote the number of runs of that particular model that were used in the ensemble.

We ranked third on the task leader board with a test score of 0.810. The top-performing team has a score of 0.823, while the second team has a score of 0.814.


There is no education like adversity .
The way to right wrongs is to turn the light of truth upon them .
April showers bring May flowers
Never stop dreaming .

Model's Output

There is no education like adversity .
The way to right wrongs is to turn the light of truth upon them .
April showers bring May flowers
Never stop dreaming .

Ground Truth

Figure 3: Heatmap of emphases

Figure 3 shows some examples, with a heatmap showing the XLNet_Large + Classifier model's predicted score and ground truth probabilities. The model performs well for the first two examples, where the prediction is fairly accurate for all the words. For the last two examples, the model fails for some of the words. In general, the model ceases to perform well on short sentences with three to four words.

6 Conclusion

We described the systems used for submission in the Task of Emphasis Selection For Written Text in Visual Media. The task was similar to the Sequence Labeling task, and hence, similar approaches can be used for this task. Our main approach used Transformer-based models like BERT, RoBERTa, XLNet, and their ensembles. These models are pre-trained and hence, perform better after fine-tuning on our small dataset.

Future work includes making an application that automates the process of poster or advertisement making. A short written text will be the input to the app. It then predicts the probability of various words to be emphasized based on predictions from our model and also adds an appropriate image as the background either using GANs (Radford et al., 2015) or using some other API. Finally, it will output a poster, flyer, or advertisement.

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

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

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

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

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*.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Amirreza Shirani, Franck Dernoncourt, Paul Asente, Nedim Lipka, Seokhwan Kim, Jose Echevarria, and Thamar Solorio. 2019. Learning emphasis selection for written text in visual media from crowd-sourced label distributions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1167–1172.

Amirreza Shirani, Franck Dernoncourt, Nedim Lipka, Paul Asente, Jose Echevarria, and Thamar Solorio. 2020. Semeval-2020 task 10: Emphasis selection for written text in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.