FPAI at SemEval-2020 Task 10: A Query Enhanced Model with RoBERTa for Emphasis Selection

Chenyang Guo[*]**, Xiaolong Hou**[*] **, Junsong Ren, Lianxin Jiang, Yang Mo,**
Haiqin Yang, Jianping Shen
Ping An Life Insurance, Lt
ShenZhen, China
{guochenyang210, houxiaolong430, renjunsong941,
jianglianxin769, moyang853, yanghaiqin260,
shenjianping324}@pingan.com.cn

Abstract

This paper describes the model we apply in the SemEval-2020 Task 10. We formalize the task of emphasis selection as a simplified query-based machine reading comprehension (MRC) task, i.e. answering a fixed question of "Find candidates for emphasis". We propose our subword puzzle encoding mechanism and subword fusion layer to align and fuse subwords. By introducing the semantic prior knowledge of the informative query and some other techniques, we attain the 7th place during the evaluation phase and the first place during train phase.

1 Introduction

The SemEval-2020 Task 10 focuses on the emphasis selection in visual text media, i.e. choosing candidates for emphasis in short written text, to enable automated design assistance in authoring (Shirani et al., 2019).

In this paper, we propose a novel model enhanced by an informative query with RoBERTa (Liu et al., 2019) as the backbone to select the emphasis in a sentence. We are inspired by Li et al. (2019) who introduced a MRC approach to improve the performance of the NER task by querying differently for diverse named entities. We reform the MRC query and formalize the emphasis selection task as a simplified question answering task in machine reading comprehension (MRC). That is, given a natural language query, emphasis tokens are extracted to answer the query. For example, the task of selecting emphasis tokens "dangerous", "evolve", "not", "to" in "What 's dangerous is not to evolve" is formalized as an MRC task of answering a fixed query, "Find candidates for emphasis". Additionally, we propose subword puzzle encoding and the subword fusion layer to solve the token alignment problem gained from RoBERTa. Our work shows that the informative query is beneficial in the case of emphasis selection.

2 Background

The purpose of this task is to design automatic methods for selecting emphasis on visual media such as flyers, posters and ads. The selected emphasis, written in English, is highlighted to grab a viewer's attention. The competition dataset consists of 3,143 sentences with labels and 743 sentences without labels. Details about the dataset are introduced in (Shirani et al., 2019) and (Shirani et al., 2020). Most of sentences in the dataset contain a few words, usually less than 20.

3 Model

Figure 1 depicts our proposed model: given a sentence, we apply concatenation to the uniform query and the sentence to obtain the input. Then we apply subword puzzle encoding to store the affiliation information of subwords. After that, we apply RoBERTa to get the representation of each token, i.e. the embeddings from the last two hidden layers in RoBERTa. They are then fused in the subword fusion layer and output by a fully connected layer to produce regression score.

[*] These authors contributed equally to this work

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1652–1657
Barcelona, Spain (Online), December 12, 2020.

The model is fed with chunks of sentences and a uniform query is combined with each sentence in a sentence pair format[1] at the entrance. Then it is processed by the subword puzzle and fed into RoBERTa for encoding.

$$[L_1, L_2, ..., L_{12}]^T = RoBERTa(q_1, q_2, ..., q_M; x_1, x_2, ..., x_N), \tag{1}$$

where $L_k \in \mathbb{R}^{\hat{N} \times E}, k = 1, 2, ..., 12$, denotes the embeddings from each transformer unit within RoBERTa, the length of subwords $\hat{N}$ is usually not less than N due to its generation mechanism by Byte-Pair Encoding (BPE) (Sennrich et al., 2015), E refers to the embedding size of RoBERTa and is set to 768 as default. $q_i, i = 1, 2, ..., M$, denotes M tokens in the query and $x_j, j = 1, 2, ..., N$, denotes maximum length of N tokens in the original sentence (sentences which are shorter than N will be padded to length of N). We extract the embeddings of the last two layers, i.e. L_{11}, L_{12} and then pass them to subword fusion layer to produce embeddings for genuine tokens.

$$L = Fusion(L_{11}, L_{12}), \tag{2}$$

The fused embedding, $L \in \mathbb{R}^{N \times E}$ is to represent each genuine token.

In the end, the aligned embeddings are sequentially passed through a dropout layer and a full connected layer with the sigmoid activation to produce a logit score for each token.

$$\begin{aligned} L &= dropout(L), & (3)\\ Score &= Sigmoid(L \cdot W + b). & (4)\end{aligned}$$

where $Score \in \mathbb{R}^{N \times 1}, W \in \mathbb{R}^{E \times 1}, b \in \mathbb{R}^{N \times 1}$, the score represents the emphasis of each word.

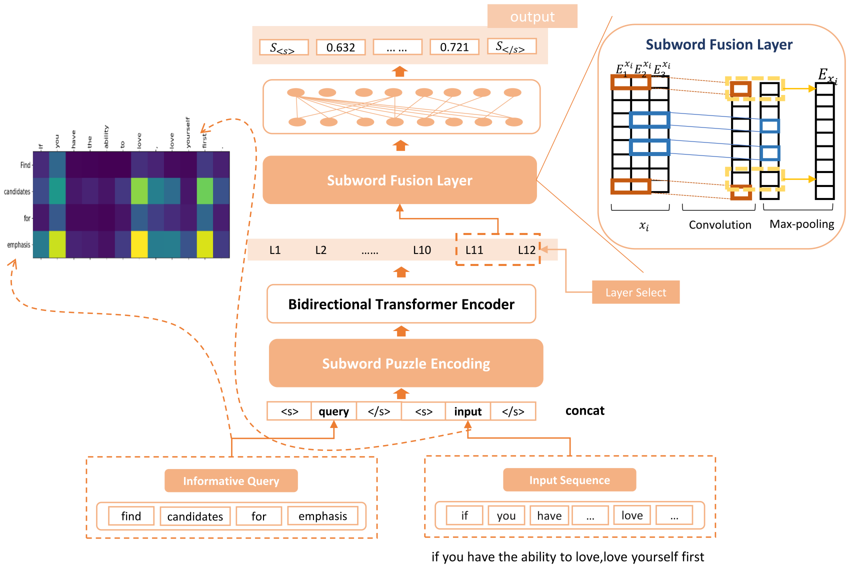

Figure 1: Query Enhanced Model

3.1 Informative Query

We introduce an extra query to help the model understand its goal. The model can benefit from the query since that key words in the query can draw attention, help model focus on specific features of a word and embed **semantic prior knowledge** into it. Prior knowledge from key words can also accelerate training and it is especially useful when fine-tuning a pre-trained language model, since the language model has the key word embeddings within itself.

[1] The sentence pair format in RoBERTa is $< s >$ sent 1 $< /s >< s >$ sent 2 $< /s >$

3.2 Subword Puzzle Encoding

Byte-Pair Encoding (BPE) and Sentencepiece (Kudo and Richardson, 2018) are two effective subword techniques to relieve the Out-of-Vocabulary (OOV) problem. We follow the implementation in RoBERTa and XLNet (Yang et al., 2019) in our work. The forward processes of these language models will yield a sequence longer than the original one. This is because those words out of vocabulary will be cut into no less than one subword. Hence we have to align the output sequence with the actual sentence token sequence.

We observe that simply encoding each token in a sentence each time will generate a different subword sequence from encoding the whole sentence. It is in that BPE will depend on the context of a word to split it. Therefore we propose subword fusion layer to overcome this problem. Firstly, we have to know the affiliation, i.e. which subwords the real token owns. Separately encoding each word cannot show the correct indexes when words occur together in a sentence, so we propose a new mechanism called **puzzle encoding**.

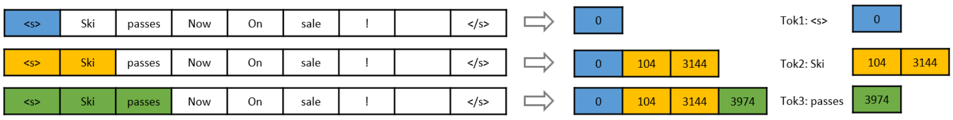

Figure 2: Subword Puzzle Encoding

For each time, we add one next word and encode words together with the preceding words, we cut the encoded sequence puzzle from the previous one. Then we acquire the newly extended ids as the subwords that belong to this newly added word.

3.3 Subword Fusion Layer

In this layer, we fuse the embeddings extracted from multiple layers in RoBERTa, utilizing affiliation information given by puzzle encoding. We show two approaches of fusion in the following.

Average: Suppose the subwords have the same contribution to the whole word's embeddings, we average all subwords' embeddings to represent the word. Let $x_i, i = 1, 2, ..., N$ denotes words within a sentence, E_{x_i} denotes the embeddings of x_i, $E_j^{x_i}$ denotes the jth embeddings of x_i, N_{x_i} denotes the number of x_i's subwords

$$E_{x_i} = \frac{1}{N_{x_i}} \sum_{j=1}^{N_{x_i}} E_j^{x_i}. \tag{5}$$

CNN Extraction: This method is inspired by Li et al. (2018). The concept is that subwords have N-gram information related innerly, so a convolution layer with kernel-size of $1 \times N$ is trained to extract features and cast to several new vectors which enjoy same dimensions as the subwords' embeddings. After applying the max-pooling to pool out the greatest value for each dimension, we attain the corresponding vector to represent the word. Figure 3 illustrates an example on CNN extraction.

4 Experiments

In the following, we show the parameters we used for replication and discuss how to select hidden layers and queries. We show the improvement using pseudo labels and deliver an introduction about an attempt on a man-revision.

4.1 Parameter Settings

After comparison and experiments, we select a fixed query **"Find candidates for emphasis"** for this task. The query is concatenated with each original sentence before passing through our system. The **average** embeddings of all subwords belonging to the word are applied for the subword fusion.

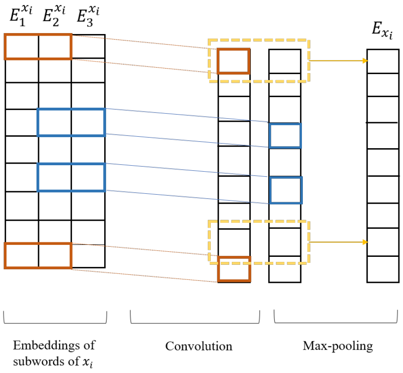
$E_1^{x_i}$ $E_2^{x_i}$ $E_3^{x_i}$
E_{x_i}
Embeddings of subwords of x_i
Convolution
Max-pooling

Figure 3: An example on CNN extraction: suppose x_i has 3 subwords, utilize a kernel size of 1×2

Query	Score
Find candidates for emphasis	0.800
Choose candidates for emphasis	0.797
Choose the first four candidates for emphasis	0.794
Find the first four candidates for emphasis	0.792
Emphasis	0.792
Null	0.790
Emphasis is the strengthening of words in a text with a font in a different style from the rest of the text	0.785

Table 1: Scores of different queries

Kullback-Leibler divergence loss (Kullback and Leibler, 1951) is adopted as the loss function. The Adam optimizer (Kingma and Ba, 2014) is applied with $lr = 1e - 5, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e - 8$. We deliver 10-fold cross validation on the training data. Besides, we append a copy of pseudo labels to each fold of training data and train on **softmaxed labels**[2] instead of the original ones. By setting the max length to $N = 50$, batch-size to 32, epoch to 10, we obtain 10 models trained on different folds of data and we average the result of the 10 models as our final prediction.

4.2 Hidden Layer Selection

We search the best combination of hidden layers from RoBERTa. There are 3 ways to merge multiple layers, average, concatenation and max pooling. There are 12 transformer units in RoBERTa and we observe that **average on the** 11^{th} **and the** 12^{th} **layer** performs the best, which implies that the upper layers capture more information than the lower layers. It proved that after being forward processed by many transformers units, the word representation absorbs more information for our downstream task.

Layer	**Merge**	**Rank**	**Score**
11, 12	avg	1	0.7911
11, 12	max	2	0.7906
12	single	3	0.7905
9, 10, 11, 12	max	4	0.7886
11	single	5	0.7879
...	...	...	...

Table 2: Results on the upper layers

Layer	**Merge**	**Rank**	**Score**
...	...	...	...
7	single	17	0.7598
8	single	15	0.7533
9	single	18	0.7455
...	...	...	...
1, 2, 3, 4	avg	24	0.6722

Table 3: Results on the lower layers

4.3 Query Experiments

Table 1 shows the average match-m score of different queries or no query on our local cross-validation test. Most queries indeed improve the score a bit compared with the one without a query. Among queries, a task description query outperforms a single-key-word query as well as the definition for it.

Figure 4 shows the multi-head attention captured from last hidden layer. Query “Find candidates for emphasis” draws attention from original sentence. Words “love”, “you”, “first” are emphasised word from ground truth and they indeed show more attention on “candidates” and “emphasis” from the query we provide. It shows that the model indeed pays more attention to the key words from the query.

[2] Softmaxed labels compress the distance between emphasis and not emphasis in order to generate difficulties for the model, so as to improve robustness.

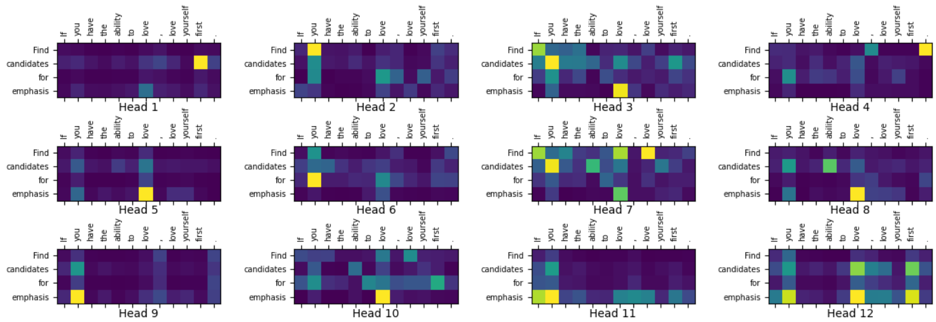

Figure 4: Multi-head attention on the query

4.4 Pseudo Label

Pseudo labeling is a semi-supervised method to augment training data (Arazo et al., 2019). We generate pseudo labels from the evaluation dataset without labels and append them to the training data. It is worth mentioning that since we have adjusted the true label to a softmaxed transformation which resulted in a more ambiguous probability distribution, the model generates pseudo labels close to that of an ambiguous probability distribution. Hence, a **reverse-softmax** transformation is required to resume the real distribution. We found models trained with pseudo labels improve around 0.005 point on average match-m score.

Reverse-Softmax:

Suppose (y_1, y_2) be softmaxed on (x_1, x_2), given (y_1, y_2),

$$y_1 = \frac{e^{-x_1}}{e^{-x_1} + e^{-x_2}}, \quad y_2 = \frac{e^{-x_2}}{e^{-x_1} + e^{-x_2}}, \quad \text{with} \quad x_1 + x_2 = 1, \tag{6}$$

and

$$x_1 = \frac{ln\frac{1-y_1}{y_1} + 1}{2}, \quad x_2 = 1 - x_1. \tag{7}$$

4.5 Man-Revision

In order to determine the effectiveness of the model, we intend to compare the discrepancy of the predicted labels and the true labels. We observe that (1) more than 80% of the misprediction is due to the position exchange of the first and the second emphasis words; and (2) the error rate of the long sentences is higher than that of short sentences.

We deliver manual revision on the predictions,focusing more on those short sentences which we consider more likely to show significant errors,but we received a 0.01 point drop on scores. We confirm that our system indeed performs quite well on this task, at least better than we do.

5 Conclusion

We reformulate the task of emphasis selection as an MRC task with a fixed query, "Find candidates for emphasis" and apply RoBERTa-base as the backbone language model. By adopting the techniques of subword fusion, pseudo labels, we attain an average score of 0.796 (match-1: 0.690, match-2: 0.780, match-3: 0.840, match-4: 0.873) and achieve the 7^{th} place in the evaluation phase and the first place in the train phase.

Our implementation shows that 1) reformulating the task into an MRC task with the subword fusion technique is effective; 2) embeddings of the last two layers of RoBERTa absorb most of the information; 3) pseudo labels can be applied to improve the model performance; 4) the man-revision does not help with the model performance improvement.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2019. Pseudo-labeling and confirmation bias in deep semi-supervised learning.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Changliang Li, Liang Li, and Ji Qi. 2018. A self-attentive model with gate mechanism for spoken language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3824–3833, Brussels, Belgium, October-November. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019. A unified mrc framework for named entity recognition.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units.

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, Florence, Italy, July. Association for Computational Linguistics.

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, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.