

# ERNIE at SemEval-2020 Task 10: Learning Word Emphasis Selection by Pre-trained Language Model

Zhengjie Huang, Shikun Feng, Weiyue Su, Xuyi Chen,  
Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, Yu Sun

Baidu Inc., China

{huangzhengjie, fengshikun01, suweiyue, chenxuyi}@baidu.com  
{wangshuohuan, liujiaxiang, ouyangxuan, sunyu02}@baidu.com

## Abstract

This paper describes the system designed by ERNIE Team which achieved the first place in SemEval-2020 Task 10: Emphasis Selection For Written Text in Visual Media. Given a sentence, we are asked to find out the most important words as the suggestion for automated design. We leverage the unsupervised pre-training model and finetune these models on our task. After our investigation, we found that the following models achieved an excellent performance in this task: ERNIE 2.0, XLM-ROBERTA, ROBERTA and ALBERT. We combine a pointwise regression loss and a pairwise ranking loss which is more close to the final  $Match_m$  metric to finetune our models. And we also find that additional feature engineering and data augmentation can help improve the performance. Our best model achieves the highest score of 0.823 and ranks first for all kinds of metrics.

## 1 Introduction

Emphasis selection for written text in visual media is proposed by Shirani et al. (2020) and Shirani et al. (2019). The purpose of this shared task is to design automatic methods for emphasis selection, i.e. choosing candidates for emphasis in short written text, to enable automated design assistance in authoring. For example, Shirani et al. (2019) mentions that such a technique can be applied to some graphic design applications such as Adobe Spark to perform automatic text layout using templates that include images and text with different fonts and colors. The major challenge is that given only thousands of annotated short text data without any context about the text or visual background images, we are asked to learn the author- or domain-specific emphatic about the short text. Besides, these short text data are annotated by crowd-sourcing workers. And we find that different annotators have different standards, which increases the difficulty of this task.

To identify the most important words, we model the task as a sequential labeling problem. Our base models leverage different unsupervised language model such as ERNIE 2.0 (Sun et al., 2019b), XLM-ROBERTA (Conneau et al., 2019), ROBERTA (Liu et al., 2019) and ALBERT (Lan et al., 2019). These large unsupervised models are pre-trained on a large amount of unannotated data and carry valuable lexical, syntactic, and semantic information in training corpora. Our approach is as follows: firstly, the word-level output representations for the sentence are computed by pre-trained models and then fed into a designed downstream neural network for word selections; secondly, we finetune the downstream networks together with the pre-trained models on the annotated training data; thirdly, we investigate several different objective functions to learn our model; and finally, we apply feature engineering and several data augmentation strategies for further improvement.

The rest of the paper is organized as follows. In Section 2, we will briefly overview some related works to our system. Section 3 shows the details of our approach. Our experiments will be shown in Section 4, and Section 5 concludes.

---

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

## 2 Related Work

Recently pre-trained models have achieved state-of-the-art results in various language understanding tasks such as BERT (Devlin et al., 2018), XLM-ROBERTA (Conneau et al., 2019), ROBERTA (Liu et al., 2019), ALBERT (Lan et al., 2019) and ERNIE 2.0 (Sun et al., 2019b). (Devlin et al., 2018) first introduced a bidirectional encoder representation from transformers called BERT. They developed several pretraining strategies such as masked language models and a next sentence prediction task. Since then, many studies about pretraining strategies come out and most of them share similar neural network architectures but with different pretraining schemes. For example, ROBERTA (Liu et al., 2019) finds that dynamically changing the masking for the masked language model and removing the next sentence prediction task can help with improving the performance in downstream tasks. ALBERT (Lan et al., 2019) adds sentence order prediction task and optimizes the memory usage of original BERT to achieve better results. XLM-ROBERTA (Conneau et al., 2019) trains its models over one hundred languages with multilingual settings and gets the first single large model for all languages.

ERNIE 2.0 (Sun et al., 2019b) is an improvement of ERNIE 1.0 (Sun et al., 2019a) and the world's first model to score over 90 in terms of the macro-average score on GLUE benchmark (Wang et al., 2018). ERNIE 1.0 (Sun et al., 2019a) introduces knowledge masking strategies. It gains a large benefit from entity-level and phrase-level masked language models. ERNIE 2.0 (Sun et al., 2019b) proposes a continual pre-training framework that incrementally builds pre-training tasks and then learns pre-trained models on these constructed tasks via continual multi-task learning. ERNIE 2.0 constructs three kinds of tasks including word-aware tasks, structure-aware tasks and semantic-aware tasks. All of these pre-training tasks rely on self-supervised or weak-supervised signals that could be obtained from massive data without human annotation. A continual multi-task learning method is proposed to improve the model's memory over different pre-training tasks.

The researchers of ERNIE 2.0 released a new version recently which made a few improvements on knowledge masking and application-oriented tasks, aiming to advance the model's general semantic representation capability. In order to improve the knowledge masking strategy, they proposed a new mutual information-based dynamic knowledge masking algorithm. They also constructed specific pre-training tasks for different applications. For example, they added a coreference resolution task to identify all expressions in a text that refer to the same entity. Details can be found on the blog<sup>1</sup>.

Following the current trends of pre-training and fine-tuning paradigm for natural language processing, our system adopts these models as our base word and sentence representation.

## 3 Our Approach

### 3.1 Word Emphasis Regression with Subword Alignment

Instead of learning label distributions with KL divergence loss like Shirani et al. (2019), we directly learn to regress the emphasis probability values with mean squared error (MSE) loss. Our model is shown in Figure 1. We simply plug in the task-specific inputs and outputs into the pre-trained model like ERNIE 2.0 does. Words are preprocessed into subword level with the WordPiece tokenizer (Wu et al., 2016) like most of the BERT-style fine-tuning tasks. The subword tokens are then fed into ERNIE 2.0 to compute the contextual representations for each subword. The task-specific output layer is a fully connected neural network with sigmoid activation to constrain the output between 0 and 1. Since our model is based on the subword level, the ground-truth scores of the words are split into pieces and each subword piece will learn its aligned emphasis value. All the parameters of ERNIE 2.0 and the final fully connected neural network will be tuned together. During the inference stage, word emphasis score will be computed by aggregating its corresponding subword scores on average as shown in Figure 1b.

### 3.2 Subword-level Pairwise Ranking Loss

Since the final metrics only consider the top 4 words with the highest emphasis scores, the individual subword level regression task ignores relative scores between the tokens and might hurt the performance.

<sup>1</sup><http://research.baidu.com/Blog/index-view?id=128>

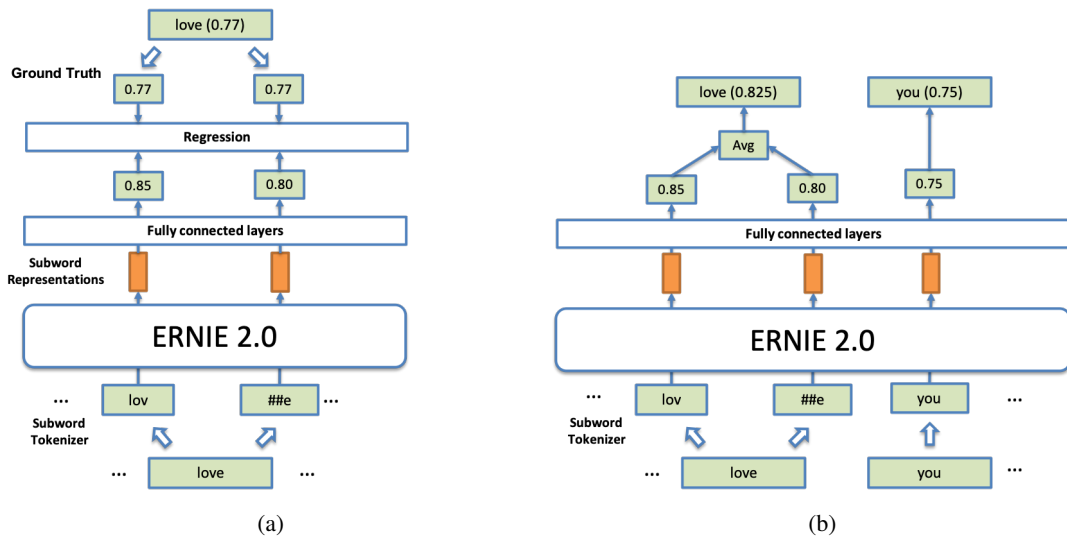


Figure 1: The input sentence will be tokenized into subword level by WordPiece algorithm. Then we use pre-trained models such as ERNIE 2.0 to get the subword representations. Next we apply a regression layer on the model as described in Figure 1a. Finally we will aggregate subword scores to get the final word emphasis like Figure 1b.

To overcome this issue, we develop a pairwise ranking loss, which considers all pairs of the subword pieces and learns the relative orders of the emphasis probability. As shown in Figure 2, the emphasis will be also split at subword level. Each subword piece will be compared with all other subword pieces that have lower scores. Then the loss is computed as follows:

$$J = \frac{1}{N^2} \sum_i^N \sum_j^N - \max(\text{score}(w_i) - \text{score}(w_j), 0) * \log \sigma(s(w_i) - s(w_j)), \quad (1)$$

where  $\sigma$  is sigmoid function and the  $\text{score}(\cdot)$  function returns the ground-truth emphasis probability labels for each subword and the  $s(\cdot)$  is the output of the logits score (without sigmoid activation). The loss is weighted by the gap of scores.

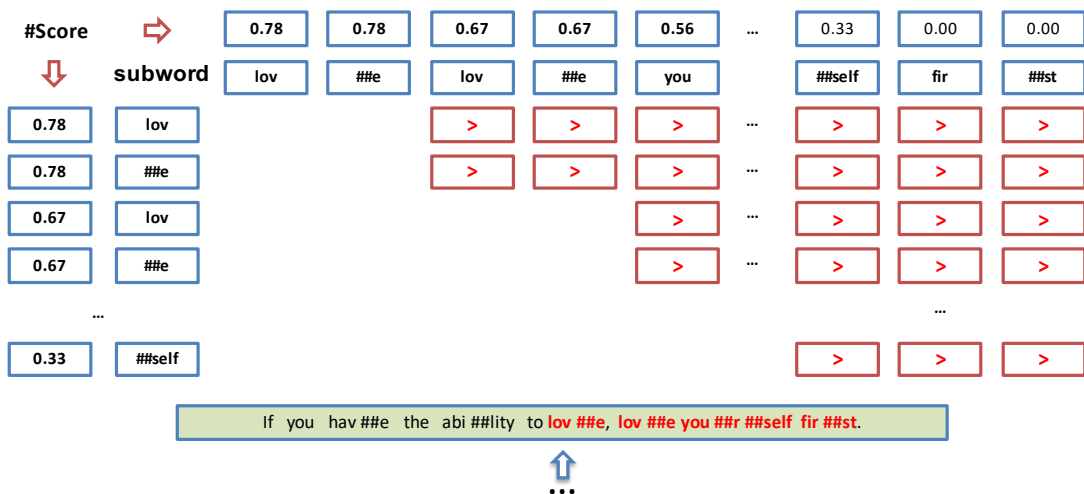


Figure 2: Pairwise ranking loss.

### 3.3 Word Emphasis Lexical Feature

After further investigating the short text data, we find that some additional features about the capitalization of words and the appearance of hashtags can bring further improvements. The statistics of the average score for different word types are shown in Table 1. Obviously, words with hashtag or capital letters are more likely to be annotated as important words. However, WordPiece algorithm separates words into pieces and drops the information about the prefix of words. For example, "#plantgang" is split into "#", "plant", and "##gang". In our model, regression loss is computed for each individual word so it is difficult for the word piece "plant" and "##gang" to capture the prefix information. Therefore, the explicit features about the special meaning of hashtag in social media and visual impact of uppercase characters are valuable. These features about the words are denoted as 0-1 vectors and concatenated with the ERNIE embedding as the inputs of the final fully-connected layers as shown in Figure 3.

Word Types	Avg. Score
All	0.284
Starts with a capital letter	0.369
Word in uppercase	0.333
Starts with hashtag	0.611

Table 1: Word types and its corresponding scores.

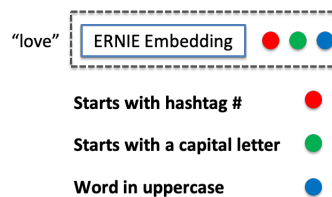


Figure 3: Additional features.

### 3.4 Data Augmentations

Because there are only 2000 annotated data, it's quite easy to overfit the training data even with pre-trained models. To enlarge the amount of annotated data, we design several data augmentation strategies as follows: 1) randomly remove a word, 2) randomly uppercase a word, and 3) randomly reverse a sentence. We find that the augmentation can help delay the overfitting occurrence, especially for large models. The details of the augmentation schemes are shown in Table 2. Each scheme is triggered independently for each word and sentence with the given probabilities. Therefore, we can have many different modified versions of the origin sentences to delay the phenomenon of overfitting.

Augmentation Schemes	Probability
Randomly remove a word	1%
Randomly uppercase a word	5%
Reverse the sentence	10%

Table 2: Data augmentation settings.

## 4 Experimental Results

All experiments are executed on an Nvidia V100 GPU. Each model runs 10 epochs with early stopping strategies based on the performance in the validation set. Since the training and validation data are both small, we find that models have a large variant performances among each run.<sup>2</sup> So we combine all the provided training and validation sets, then split them in a random 8-Fold settings. For each model in each fold, we run five times and we will report the average score on our 8-Fold settings to get a much more stable analysis.

Table 3 shows the score across different models, we find that ERNIE 2.0 is the most powerful base model among several different pre-trained models, and gets 0.781 average ranking score over 8-Fold cross-validation.

Table 4 shows the score gains across different fine-tuning strategies. We report the average gain and the maximum gain over the average score for each base model on the 8-Fold settings. We find that not all

<sup>2</sup>The performance on different fold can have a large variance different. For example, XLM-ROBERTA-LARGE can achieve rank score from 76.7% to 80.1% in different fold runs.

Model	8FoldsCV-AVG. Rank
<b>ERNIE 2.0-LARGE</b>	<b>0.7810</b>
XLM-ROBERTA-LARGE	0.7791
ROBERTA-LARGE	0.7764
ALBERT-XXLARGE-v2	0.7755

Table 3: The results of different models.

Strategies	Average Gain	Max Gain
PairWise Loss	-0.00544	0.00385
<b>Lexical Features</b>	<b>0.00105</b>	<b>0.00709</b>
Data Augmentation	-0.00144	0.00344

Table 4: The score gains from different strategies.

models can benefit from these training schemes. However, they still bring large improvement to some of our models. In Figure 4, we draw the box plot of the model scores. The box plot shows that the lexical feature is the most effective strategy. Besides, data augmentation and pairwise loss can also achieve a higher score.

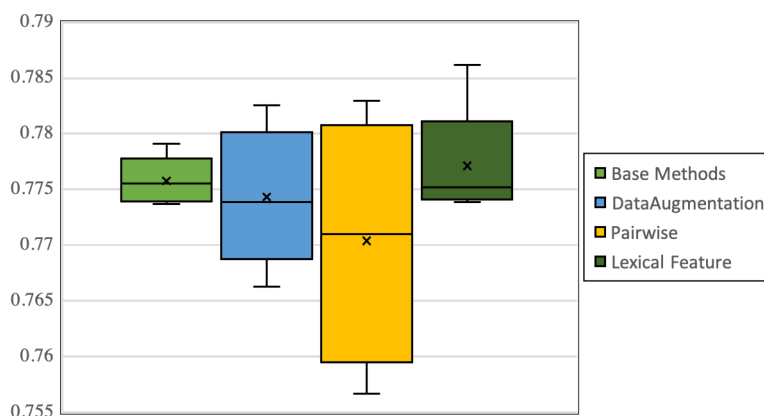


Figure 4: Validation scores among different strategies.

For final submission, we ensemble our best strategies and we achieve the highest score 0.823 ranking first for all kinds of the metrics.

## 5 Conclusion

In this paper, we present our system that ranks first in SemEval-2020 Task 10. Our solution contains several strategies and we provide detailed experiments to analyze which of them are effective. Our experiments show that models empowered by pre-trained language models are most effective, especially for ERNIE 2.0. Besides, lexical features, pairwise loss, and data augmentation can also bring improvement for some of our models.

## References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- 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*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- 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*.
- Amirreza Shirani, Franck Dernoncourt, Paul Asente, Nedim Lipka, Seokhwan Kim, Jose Echevarria, and Tamar 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 Tamar Solorio. 2020. Semeval-2020 task 10: Emphasis selection for written text in visual media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019b. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- 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. *arXiv preprint arXiv:1609.08144*.