YNU-HPCC at SemEval-2020 Task 10: Using a Multi-granularity Ordinal Classification of the BiLSTM Model for Emphasis Selection

Dawei Liao, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact:davidblackingongan@qq.com
{wangjin, xjzhang}@ynu.edu.cn

Abstract

Written text emphasis in visual media is used to increase the comprehension of written text, to grab a viewer's attention, and to convey the author's intent.The task is choosing candidates for emphasis in short written text, to enable automated design assistance in authoring. As the author's intent is unknown and only the input text is available, multiple emphasis selections are valid. In this study, we propose a multi-granularity ordinal classification method to address the problem of emphasis selection. Specifically, word embeddings are learned from the Embeddings from Language Model (ELMo) to extract feature vector representations. Then, the ordinal classifications are implemented on four different multi-granularities to approximate the continuous emphasized values. Comparative experiments were conducted to compare the model with the baseline, in which the problem is transformed to a label distribution problem.The code of this paper is availabled at: https://github.com/DavidInWuhanChina/SemEval-2020-Task10.

1 Introduction

Short texts have a great impact on visual communication. They are usually designed to grab a viewer's attention and convey a message efficiently. For text, word emphasis is used to capture the intent better, removing the ambiguity that may exist in some plain texts. Word emphasis can clarify or even change the meaning of a sentence by drawing attention to some specific information, and it can be done with colors, backgrounds, fonts, italics, or boldface. The purpose of this 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 (Shirani et al., 2020). In the task definition, given a sequence of words or tokens, there is a subset of words that are good candidates to emphasize, and the word probability represents the degree of emphasis, which needs to be predicted.

In a previous work, a rule-based approach is used (Widera et al., 1997). Later, Text-based features such as part-of-speech (POS), information content, position in the sentence and other information was adopted (Volker Strom, 2007). Some methods have been proposed for predicting emphasized words for expressive Text-To-Speech (TTS) based on a deep neural network (DNN) (Mass et al., 2018; Rosenberg et al., 2015). To address the multiple annotators problem, a majority voting ensemble has been used to transform the problem into single-label learning (Laws et al., 2011).

Shirani (2019) suggest that the task should be transformed into label distribution learning (LDL). The main difference between such a method and previous works is that the label is not a single dispersed value, but rather a continuous label distribution. Therefore, the model can fit more accurately to a real label instead of having the distortion of transforming a probability into a single label.

The essence of the problem is to use a sequence labeling model to predict a probability value for each word. The existing methods mainly applied either a softmax or conditional random field (CRF) (Huang et al., 2015; Lafferty et al., 2001) layer to predict labels, which cannot be used in this task. Therefore, using a linear decoder to output continuous linear values or using sigmoid output nonlinear probability values as

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1710–1715
Barcelona, Spain (Online), December 12, 2020.

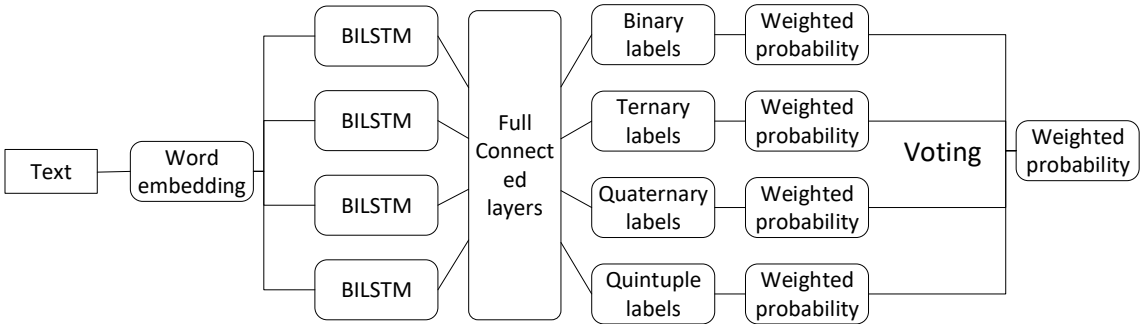


Figure 1: Overview of our model.

the result have been considered, but the performance was not suitable.

In this paper, we propose a multi-granularity ordinal classification method for the task. This involves an ensemble of four different models with different levels of granularity, including Binary Classification, Ternary Classification, Quaternary Classification, and Quintuple Classification. Specifically, it divides the probability between (0,1) into different parts as category labels. For the Binary Classification, the probability (0,1) is divided into two categories, i.e., (0-0.5) and (0.5-1). For the Ternary Classification, the probability (0,1) is divided into three categories, i.e., (0-0.33), (0.33-66), and (0.66-1). For the Quaternary Classification, the probability (0,1) is divided into four categories, i.e., (0-0.25), (0.25-0.5), (0.5-0.75), and (0.75-1). The Quintuple Classification divides (0,1) into five categories, i.e., (0-0.2), (0.2-0.4), (0.4-0.6), (0.6-0.8), and (0.8-1). The reason why we use multi-granularity ordinal classification is that Binary Classification ignores the differences between word emphasis probability. For example, if two words' emphasis probabilities are 0.3 and 0.4, they belong to the same class for Binary Classification, while they belong to different classes for ternary classification. The finer the granularity of the division is, the more the model can learn the difference between key words. However, there is still a balance between granularity and the number of classifications, that is, the performance of multi-granularity classification will decrease as the granularity increases. This approach can use different granularities of information to improve performance.

The rest of the paper is organized as follows. In Section 2, we describe the multi-granularity ordinal classification of the bidirectional long short-term memory (BiLSTM) model in detail. The comparative experimental results are presented in Section 3. Conclusions are drawn in Section 4.

2 Multi-granularity Ordinal Classification of BiLSTM Model

We use an ensemble of multi-granularity ordinal classification of the BiLSTM model to learn emphasis patterns. Figure 1 shows the overall architecture of the proposed model. First, we fine-grain words' emphasis probabilities, and the granularity labels (0, 1, 2, 3, 4) denote the emphasis degrees from high to low. After we load the test into the word embedding layer, a BiLSTM is used to obtain word annotations that summarize the information from both directions and to learn more abstract features. A fully connected softmax layer is used to output a probability distribution over all classes from the BiLSTM output, which can be weighted into a single-value probability. Finally, we use a voting ensemble to obtain the final probability.

2.1 Multi-granularity Ordinal Classification

Multi-granularity ordinal classification is widely applied in many domains. Multi-granularity classification is used to analyze visual objects from subordinate categories, e.g., species of birds or models of cars in computer vision (Wei et al., 2019; Berg et al., 2014). Multi-granularity classification is also used in many NLP tasks such as sentiment classification (Hao et al., 2019), neural machine translation (Mehri and Eskénazi, 2019), Dialog (Mehri and Eskénazi, 2019)and named entity recognition (Mai et al., 2018).

In this task, there are two methods of fine-graining, ROC-AUC and ORDER.

ROC-AUC: the probability of each word between (0,1) is divided into different parts as category labels

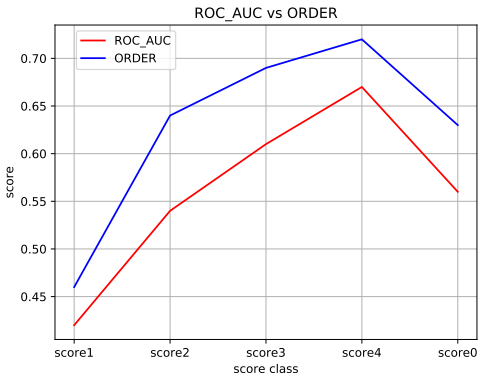


Figure 2: The comparison between ROC-AUC and ORDER in Binary Classification.

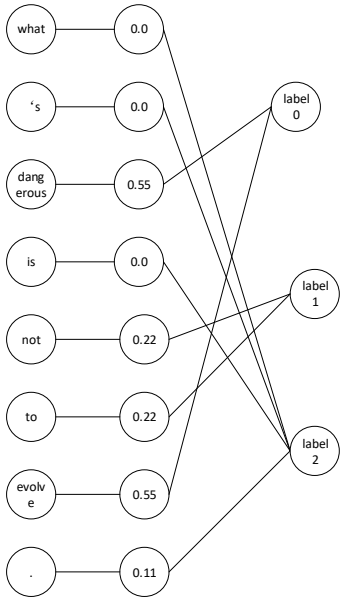


Figure 3: ORDER in Ternary Classification.

(0, 1, 2, 3, 4), representing the degree of emphasis. Experimental results show that when the threshold is set to 0.35, the model performs best in Binary Classification.

ORDER: First, we arrange the words according to probability from large to small, and then take the previous k words into a category, after which we take the remaining m words into a category, and so on until the specified granularity is obtained. We find that when k is set to 4, the model performs best in Binary Classification. Then, we compare ROC-AUC and ORDER with respect to Binary Classification. The comparison of ROC-AUC and ORDER with respect to Binary Classification is shown in Figure 2, where the score classes are defined in section 3.3.

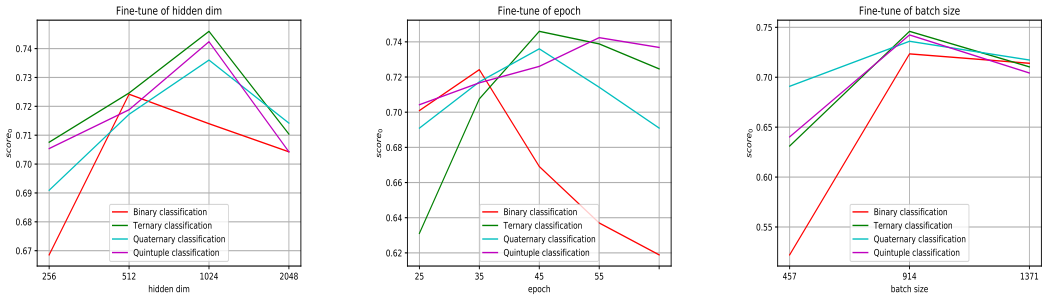
As indicated, ORDER performed better than ROC-AUC with respect to Binary Classification. Therefore, we use ORDER as our fine-graining method. Experimental results show that different partition sizes behave differently. For Binary Classification, we take the top four in probability as label 0, and the rest as label 1. For Ternary Classification, we take the top two in probability as label 0, the third to fourth as label 1, and the remaining as label 2. The division process for Ternary Classification is shown in Figure 3. For Quaternary Classification, we take the top two in probability as label 0, the third through fourth as label 1, the fifth through sixth as label 2, and the rest as label 3. For Quintuple Classification, we take the top two in probability as label 0, the third through fourth as label 1, the fifth through sixth as label 2, the seventh as label 3, and the remaining as label 4.

2.2 Word Embedding

To capture semantic and syntactic information of a word, word embedding has been widely used in the NLP domain (Lai et al., 2016).The 1024-dimensional ELMo word vector is used in the first layer of the model, and the word vector matrix is loaded into the embedding layer. Also, the max sequence length is set to 38 because 99% of sentence lengths are shorter than this value.

2.3 Bi-directional Long Short-Term Memory

Bi-directional Long Short-Term Memory (BiLSTM) is based on LSTM and is a special kind of RNN (Hochreiter, 1997), which is capable of learning long-term dependencies. Employing two BiLSTM layers helps to build a deeper feature extractor. We also find that having more than two stacked LSTM layers does not help the performance, as the model becomes too complicated.



(a) Hidden dim (b) Epoch (c) Batch size

Figure 4: Parameter selection of the proposed model evaluated on dev dataset.

	hidden dim	**batch size**	**epoch**
Binary Classification	512	914	35
Ternary Classification	1024	914	45
Quaternary Classification	1024	914	45
Quintuple Classification	1024	914	45

Table 1: The best-tuned parameters

2.4 Ensemble

The output of full connected layer is a label distribution. The output must be weighted and summed into a single value $p_k(k = 2, 3, 4, 5)$. For Binary Classification, the output of full connected layer is [a, b], where a represents the probability of label 0, b represents the probability of label 1. Compared with label 1, label 0 represents a higher probability of emphasis. So, the weighted single value $p_2 = 0.8 * a + 0.2 * b$. For Ternary Classification, the output of full connected layer is [a, b, c], and the weighted single value $p_3 = 0.44 * a + 0.33 * b + 0.23 * c$. For Quaternary Classification, the output of full connected layer is [a, b, c, d], and the weighted single value $p_4 = 0.4 * a + 0.3 * b + 0.2 * c + 0.1 * d$. For Quintuple Classification, the output of full connected layer is [a, b, c, d, e], and the weighted single value $p_5 = 0.4 * a + 0.3 * b + 0.2 * c + 0.1 * d + 0 * e$. Finally, The ensemble output $p = 0.25 * p_2 + 0.25 * p_3 + 0.25 * p_4 + 0.25 * p_5$.

3 Experiments and Evaluation

Experiments were conducted to evaluate the proposed model. We report the results of the official review. The details of the experiment are described as follows.

3.1 Data Preprocessing

The data that the organizers of the competition provided contained 6 columns, including word ID, word, begin-inside-outside (BIO) annotations, BIO frequencies, emphasis probability, and POS tags. We only used the word and corresponding emphasis probability. Then, we transformed the probability into labels using a multi-granularity ordinal classification. We obtained the max length of sentences to pad the embedding vectors and label vectors.

3.2 Implementation Details

This experiment used Keras in TensorFlow2.1. We used ELMo pretrained word vectors in tensorflow-hub. The hyperparameters were tuned to the performance of training and dev dataset using the given metric function. Different classifiers may have their own optimization parameters. For all classifiers, the learning rate is 0.001 and epsilon is 1×10^{-6} . The optimizer is RMSprop (Ruder, 2016) and loss function is mean squared error. The activation of the fully connected layer is softmax. Additional best-tuned parameters are shown in Table 1. Parameter selection for the proposed model evaluated on dev dataset is shown in

	score1	score2	score3	score4	score0
Binary Classification	0.574	0.727	0.783	0.813	0.724
Ternary Classification	0.622	0.744	0.792	0.826	0.746
Quaternary Classification	0.589	0.744	0.793	0.818	0.736
Quintuple Classification	0.607	0.751	0.790	0.822	0.742
Ensemble	0.622	0.757	0.799	0.827	0.752
dev dataset Baseline	0.592	0.752	0.804	0.822	0.742

Table 2: The dev dataset experiment results

Figure 4 .

3.3 Evaluation Metrics

For evaluation, $Match_m$ is the evaluation metric for this task: For each instance X in the test set $Dtest$, a set $S_m^{(x)}$ of m$\in$ (1,. . . , 4) words with the top m probabilities according to the ground truth. Analogously, we select a prediction set $S_m^{(x)}$ for each m, based on the predicted probabilities. The metric $Match_m$ was defined as follows:

$$\text{score}_\text{m} = \text{M}atch_m := \frac{\sum\limits_{x \in Dtext} |S_m^{(x)} \cap \hat{S}_m^{(x)}| / (\min(m, |x|))}{|Dtest|} \quad (1)$$

where $score_0$ is the average of $score_1$, $score_2$, $score_3$ and $score_4$.

3.4 Results and Discussion

We use the DL-BiLSTM+ELMo model (Shirani et al., 2019) as the baseline. The dev dataset experiment results are shown in Table 2. The results of test data in the post-evaluation period are 0.607, 0.731, 0.802, which is lower than the baseline of test data:0.608, 0.737, 0.807, 0.849, 0.75. Accroding to the dev dataset experiment results, we find that Ternary Classification performs the best, and Binary classifier performs the worst. The ensemble results are higher than any other single classification. The experimental results do not indicate that increasing the granularity of the division leads to better results. This is due to the imbalance of the data; In fact, the performance of the classifier decreases as the granularity increases.

4 Conclusion

In this paper, we describe a task system that we submitted to SemEval-2020 for emphasis selection. We propose a Multi-granularity Ordinal Classification of the BiLSTM model. In future work, we will attempt to generalize models with better capabilities.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61966038, 61702443 and 61762091. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. 2014. Birdsnap: large-scale fine-grained visual categorization of birds. In *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Multi-granularity self-attention for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, page 887–897.

Sepp Hochreiter. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991.

John Lafferty, Andrew Mccallum, Fernando C N Pereira, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Siwei Lai, Kang Liu, Liheng Xu, and Jun Zhao. 2016. How to generate a good word embedding? *IEEE Intelligent Systems*, 31:5–14, 7.

Florian Laws, Christian Scheible, and Hinrich Schütze. 2011. Active learning with amazon mechanical turk. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1546–1556.

Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Nguyen Tuan Duc, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. An empirical study on fine-grained named entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722.

Yosi Mass, Slava Shechtman, Moran Mordechay, Ron Hoory, Oren Sar Shalom, Guy Lev, and David Konopnicki. 2018. Word emphasis prediction for expressive text to speech. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2868–2872.

Shikib Mehri and Maxine Eskénazi. 2019. Multi-granularity representations of dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, page 1752–1761.

Andrew Rosenberg, Raul Fernandez, and Bhuvana Ramabhadran. 2015. Modeling phrasing and prominence using deep recurrent learning. In *Proceedings of INTERSPEECH 2015*, pages 3066–3070.

Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. *ArXiv*, abs/1609.04747.

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. 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*.

Robert A. J. Clark Yolanda Vazquez-Alvarez Jason M. Brenier Simon King Dan Jurafsky Volker Strom, Ani Nenkova. 2007. Modelling prominence and emphasis improves unit-selection synthesis. In *Proceedings of INTERSPEECH 2007*, pages 1282–1285.

Xiu-Shen Wei, Jianxin Wu, and Quan Cui. 2019. Deep learning for fine-grained image analysis: a survey. *ArXiv*, abs/1907.03069.

Christina Widera, Thomas Portele, and Maria Wolters. 1997. Prediction of word prominence. In *Proceedings of European Conf. on Speech Communication and Technology*, pages 999–1002.