Predicting Popular News Comments Based on Multi-Target Text Matching Model

Deli Chen[1], Shuming Ma[1⋆], Pengcheng Yang[1,2], and Qi Su[3]

[1] MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[2] Center for Data Science, Beijing Institute of Big Data Research, Peking University
[3] School of Foreign Languages, Peking University
{chendeli,shumingma,yang_pc,sukia}@pku.edu.cn

Abstract. With the development of information technology, there is explosive growth in the number of online comment concerning news, blogs and so on. Good comments can improve the experience of reading, but the massive comments are overloaded, and the qualities of them vary greatly. Therefore, it is necessary to predict popular comments from all the comments. In this work, we introduce a novel task: popular comment prediction (PCP), which aims to find out which comments will be popular automatically. First, we construct a news comment corpus: Toutiao Comment Dataset, which consists of news, comments, and the corresponding label. Second, we analyze the dataset and find the popularity of comments can be measured in three aspects: informativeness, consistency, and novelty. Finally, we propose a novel multi-target text matching model, which can measure these three aspects by referring to the news and surrounding comments. Experimental results show that our method can outperform various baselines by a large margin on the new dataset.

Keywords: Application · News Comment · Deep Learning

1 Introduction

With the development of information technology, more and more people begin to express their opinions on the Internet, leading to explosive growth in the number of online comment concerning news, blogs and so on. Good comments can improve the reading experience of users by showing others' attitudes and thoughts. However, it is obvious that the comments generated by lots of users are overload and the qualities of them vary greatly. So it can be very valuable to predict which comments are popular and present them with news together. This method can be beneficial to both news readers, and providers for it can improve users reading experience and increase user loyalty.

In this paper, we explore how to automatically predict the popularity of online comments based on their text data and the relevant auxiliary information, which we call the task of popular comment prediction (PCP). The popularity of the comments is influenced by a variety of factors. For instance, the quality of the comment itself, the relation between the comment and the topic of news. This leads to a fundamental question: what are the crucial aspects that characterize a popular comment? To finding out

⋆ The first two authors make equal contribution.

Table 1. An example of Toutiao Comment Dataset. The original text in the dataset is in Chinese, so we give the translation of the text. And for each comment, we show the likes number and replies number.

Title	国家车辆选号系统遭受黑客攻击。 The national vehicle license plate selection system gets hacked.
Abstract	出人意料的是黑客攻击了国家车辆选号系统，他们使用这一系统获得了很多有着好的号码的车牌，并且出售这些车牌以牟利。 It is beyond our imagination that hackers invade the national vehicle license plate selection system. They use the system get many plates of good number, and then sell them for profit.
Body	为什么那些有着好的号码的车牌那么难以获得？选择系统存在着什么问题么？... Why are those vehicle license plates with good number are hardly to get? Is there anything wrong with the selection system...
Type	社会　Society
Comment #1	车辆号牌能够买卖使我觉得搞笑,政策不是规定禁止车牌买卖吗? It makes me feel funny that the vehicle license plate can be sold or bought. Isn't it forbidden by the policy? (247 Likes, 3 Replies)
Comment #2	前排抢沙发！ I am the first one to make a comment! (0 Likes, 0 Reply)

the question, we collect some user's opinions of which factor influence the popularity of comment by questionnaire survey with the sample size of 50. We collected and analyzed the result and finding out that the factors focus on the following three aspects:

- **Informativeness:** A popular comment is usually informative and contains sufficient useful information.
- **Consistency:** A popular comment is usually highly consistent with the corresponding topic, which is decided by the news.
- **Novelty:** A popular comment tends to be novel and able to stand out from a large number of comments.

The measurements for consistency and novelty are about two parts of texts (comment and news, comment and surrounding comment). So in this view, the PCP can be seen as a subtask of Natural Language Sentence Matching (NLSM). However, different from the traditional sentences matching tasks, such as answer selection and paraphrase identification, which usually contain two parts of texts. In PCP task, we need to consider the matching between comment and different kinds of auxiliary information jointly.

So we propose the Multi-Target Text Matching (MTTM) model, which can automatically assess the popularity of online comments by referring to the relevant auxiliary information including news title, news abstract, and surrounding comments. More specifically, our model measures the informativeness of a comment by the comment itself, the consistency by matching the comments with the news, and the novelty by referring to the surrounding comments. Experimental results show that our model's scoring is highly correlated with human scoring in all of the aspects.

It is a big challenge that we lack annotated dataset for news comments. Moreover, we need comments' popularity label to conduct a supervised method. In this work, we propose the Toutiao Comment Dataset for this task. It contains the user-generated

information that can be used as the popularity label of the comment. The details will be introduced in the next section. The contributions of this paper are listed as follows:

- We propose the task of popular comment prediction (PCP), and construct a large-scale annotated dataset.
- We find three metrics which can measure the popularity of comments: informativeness, consistency, and novelty.
- We propose Multi-Target Text Matching model (MTTM), which can consider all the three metrics to predict the popularity of comments. Our model outperforms various baselines by a large margin.

2 Toutiao Comment Dataset

In this section, we introduce the proposed Toutiao Comment Dataset. The existing comment datasets, such as SFU Opinion and Comments Corpus [7], do not contain the annotated information, so they are not suitable in this task. Therefore, we construct Toutiao Comment Dataset, which contains both news and comments. More importantly, the dataset contains annotated popularity information, i.e. the number of likes, which is naturally generated by users.

Table 2. Statics information of the textual attributes (Avg-word and Avg-char denote the average number of words and characters, respectively. Vocab means the vocabulary size).

Attribute	Avg-Word	Avg-Char	Vocab
Title	16.64	24.02	36378
Abstract	75.95	114.24	46533
Body	326.17	523.78	63425
Comment	18.37	25.67	53916

Table 1 shows an example of our data. Each piece of data has five attributes: title, abstract, body, type, and a list of comments, and each comment has associated numbers of likes and replies. Table 1 also shows two examples of comments.

Users will click the likes button if they appreciate the comment, so we suppose that comment with more likes will be a popular one. To prove this, we annotated the popularity score(from 1 to 5) for three hundred comments and conducted the Pearson correlation test for the human score, and the comment likes number. The result shows that they are highly correlated(Pearson correlation coefficient is 0.82 and p-value is 0.023). However, there still exists the risk that the comment with more likes could be discriminative or offensive. So we conduct a manual sampling inspection on our dataset(count two thousand comments with more than ten likes), and the result shows that the evil items in our dataset account for only a very small proportion(smaller than 1%). So we think it is reasonable to use the likes number as the natural measurement of comment popularity. We annotate the comments whose number of likes is more than ten as popular comments and the rest as common comments.

Table 2 presents some statistics of the dataset. The average number of words in one comment is 18.37, which is close to the title (16.64). However, the vocabulary

Table 3. Numbers of examples of different classes (common comment and popular comment) in different sets.

Class	Train	Valid	Test	Total
Common	165,423	4,287	4,772	174,482
Popular	197,331	5,713	5,228	208,272
Total	362,754	10,000	10,000	382,754

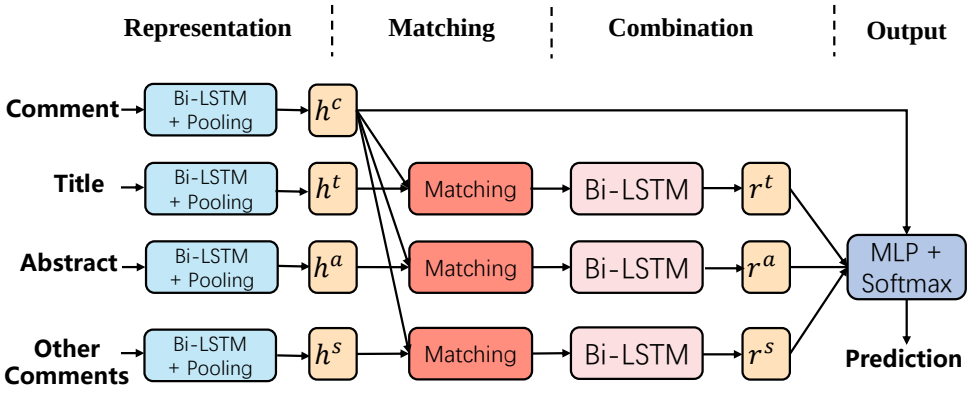

Fig. 1. The overview of the proposed MTTM model.

size of comment (53,916) is much larger than the title (36,378). The reason is that the expression in the user-generated comments is more informal and diverse. As shown in Table 2, the average length of the news body is 326.17, which is too long to be represented by general neural networks. Moreover, the abstract contains the main idea of the news, so we use the abstract instead of the news body to capture the content information.

We divide the dataset into training, validation and test sets. Both the number of samples in the validation set and test set are 10,000, and the number of samples in the train set is 362,754. The numbers of examples of different classes in different sets are shown in Table 3. The Toutiao Comment Dataset will be released soon. It is large and includes news, comments and the corresponding label, so we think it can also be used in other studies about news comment.

3 Proposed Model

3.1 Problem Formulation

Here, we give the notations and the formulation of the task. Suppose we have a set of N example in dataset $\{x_1, x_2, \cdots, x_N\}$, and each example contains a title, an abstract, a comment,and several surrounding comments: $\boldsymbol{x} = \{\boldsymbol{t}, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{s}\}$. Each comment has a label l of whether the comment is of high-popularity or low-popularity. Our goal is to assign the popularity label for each upcoming comment.

3.2 Overview

In order to predict the popularity label l, the proposed MTTM (Multi-Target Text Matching) model estimates the probability distribution $P(l|\boldsymbol{x}) = P(l|\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{a}, \boldsymbol{s})$. In our model, the popularity of a news comment can be measured using three aspects: informativeness, consistency, and novelty. The informativeness is assessed by the comment itself. The consistency is evaluated by referring to the title and the abstract. Moreover, the novelty is assessed by comparing the comment with the surrounding comments. Our model takes consideration of these aspects and gives a general justification to the popularity of the comment. More specifically, our model first represents the comments, titles, and the abstract into vectors with the Bi-LSTM [3]. Then the vectors are fed into a mean-pooling layer and becomes text-level representations. After that, the representations of the comments are matched with the titles, abstracts, and the surrounding comments respectively. The combination layer is used to combine these three aspects, and the output layer finally predicts the popularity label. The overview of the proposed model can be found in Figure 1.

3.3 Multi-Target Text Matching Model

We now give a detailed explanation of each component. Our model consists of the following four layers:

1. Representation Layer: The representation layer is to represent the comments, titles, and abstracts with dense vectors. It first transforms the words into word vectors $\boldsymbol{e} = \{e_1, e_2, \cdots, e_L\}$ (L denotes the number of words). Then, the word vectors are fed into a Bi-LSTM to obtain the forward context representation. We show the formula for the comment $\boldsymbol{c}$ as example:

$$\boldsymbol{s}^c = \mathrm{BiLSTM}_1(\boldsymbol{e}^c) \tag{1}$$

After getting the word-level representations, we use a mean-pooling layer to catch the n-gram information. We apply the overlapping mean-pooling layer to the hidden states in every time-step of Bi-LSTM. We calculate the average of the adjacent ps hidden states and the stride is 1. The size of the mean-pooling ps is a hyperparameter. The experimental results show that this is helpful to improve the performance of the model.

$$\boldsymbol{h}_i^c = \frac{\sum_{k=0}^{ps-1} \boldsymbol{s}_{i+k}^c}{ps} \tag{2}$$

where $i = 1, 2, ..., L - ps + 1$. The similar computation is performed to obtain the representations of titles $\boldsymbol{h}^t$, abstracts $\boldsymbol{h}^a$ and other comments $\boldsymbol{h}^s$.

2. Matching Layer: The matching layer uses attention mechanism to measure the similarity between the comment and the title or the abstract. Besides, it measures the dissimilarity between the comment and the surrounding comments to assess the novelty. As is shown in Figure 1, for each hidden state in the comment representations, all hidden states in the context representations (title, abstract and surrounding comments) will be matched independently. We now take the matching between the title $\boldsymbol{t}$ and the comment

$\boldsymbol{c}$ as the example using attention mechanism:

$$\alpha_{i,j} = \boldsymbol{h}_i^c * \boldsymbol{h}_j^{t\top} \tag{3}$$

$$att_{i,j} = \frac{e^{\alpha_{i,j}}}{\sum_{j=1}^{L_t'} e^{\alpha_{i,j}}} \tag{4}$$

where $i = 1, 2, ..., L_c'$ and $j = 1, 2, ..., L_t'$. (L_c' and L_t' denote the number of hidden states of comment and title's hidden states after pooling, respectively.) Then, we take $\alpha_{i,j}$ as the weight of $\boldsymbol{h}_j^t$, and access an attentive vector for the entire title $\boldsymbol{t}$ by weighted summing all the $\boldsymbol{h}_j^t$:

$$\boldsymbol{h}_i^{t_{sum}} = \sum_{j=1}^{L_t'} \boldsymbol{h}_j^t * att_{i,j} \tag{5}$$

After getting the weighted-sum vectors, we perform the matching operation:

$$\boldsymbol{mt}_i^k = f_m(\boldsymbol{h}_i^c, \boldsymbol{h}_i^{t_{sum}}, W^k) \tag{6}$$

Where $i = 1, 2, ..., L_c'$ and $k = 1, 2, ..., p$, p is the number of perspectives [14]. And the f_m is defined in the following way:

$$f_m(v_1, v_2, W) = \cos(\boldsymbol{v_1} \circ W, \boldsymbol{v_2} \circ W) \tag{7}$$

The $\circ$ is the element-wise multiplication and the W is the parameter matrix. Finally, we get the matching vectors for the title from different perspectives.

$$\boldsymbol{mt_i} = [mt_i^1, mt_i^2, ..., mt_i^p] \tag{8}$$

where $i = 1, 2, ..., L_c'$. Now we get the matching result vector between title and comment: $\boldsymbol{mt_i}$. $\boldsymbol{mt_i}$ is the matching result for one time-step, so we connect all the time-steps' results and get the matching results $\boldsymbol{mt}$ for the whole sentences. We can also get other matching result $\boldsymbol{ma}$ (matching with abstract), $\boldsymbol{ms}$ (matching with other comments) by the same way.

3. Combination Layer: The combination layer is to combine different components of matching vectors into a vector for prediction. In our model, the popularity of news comments can be measured from three aspects: informativeness, consistency and novelty. The informativeness is represented by the mean-pooling of comment's representation.

$$\boldsymbol{R}_{info} = \frac{\sum_{i=1}^{L} \boldsymbol{s}_i^c}{L} \tag{9}$$

In the previous layer, we get the matching result $\boldsymbol{mt}$. Here we use another Bi-LSTM to process the matching result:

$$\boldsymbol{S^t} = \mathrm{BiLSTM}_2(\boldsymbol{mt}) \tag{10}$$

After this, we choose the last time-step of $\boldsymbol{S^t}$ from both directions to form the vector $\boldsymbol{r^t}$ for prediction. Similarly, we get $\boldsymbol{r^a}$ and $\boldsymbol{r^s}$. The consistency is measured by the matching result between comment and title(abstract). And the novelty is directly measured by

the matching result between comment and surrounding comments.

$$\boldsymbol{R}_{cons} = [\boldsymbol{r}^{\boldsymbol{t}}, \boldsymbol{r}^{\boldsymbol{a}}] \tag{11}$$

$$\boldsymbol{R}_{nove} = \boldsymbol{r}^{\boldsymbol{s}} \tag{12}$$

Then we just connect all this three parts and get the final vector for prediction.

$$\boldsymbol{R} = [\boldsymbol{R}_{info}, \boldsymbol{R}_{cons}, \boldsymbol{R}_{nove}] \tag{13}$$

4. Output Layer: The out layer is to evaluate the probability distribution $P(l|\boldsymbol{t}, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{s})$ and output the prediction of comment label . In this layer, we simply use three layer feed-forward neural network to predict the result.

$$p(l|\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{a}, \boldsymbol{s}) = \mathrm{softmax}(\boldsymbol{W_o R} + \boldsymbol{b_o}) \tag{14}$$

4 Experiments

4.1 Experimental Details

We adopt the accuracy and macro-F_1 score as our evaluation metrics. The word embedding with 200 dimensions is initialized using word2vec [9]. The hidden size of Bi-LSTM is 200, and the number of layers is 2. We use the Adam [5] optimizer with the initial learning rate $\alpha = 0.001$. Besides, the dropout regularization [12] with the dropout probability $p = 0.2$ is used to reduce overfitting.

4.2 Baselines

We compare our model with the following baselines(Since all the neural network baselines are designed for the matching of two texts, we match the comments and the other contexts as a whole when using them):

- **Traditional machine learning methods:** We choose several traditional machine learning classifiers, including SVM, LogisticRegression (LR), and RandomForest (RF). We use comment only for all these methods because these models can hardly handle multiple inputs.
- **Siamese-CNN (Sm-CNN):** We use the Siamese framework [2] and use CNN to get the text representation. All the texts get representations individually and then get connected for prediction. The kernel size is 3,4,5, and the kernel number is 100.
- **Siamese-LSTM (Sm-LSTM):** Similar to Siamese-CNN, the only difference is that we use LSTM to get the text representation. The hidden dimension of LSTM is 200.
- **ARC-II [4]:** ARC-II is a text matching model which improves the traditional CNN matching model by using a sliding window. This model and the following two baselines are implemented using an open-source text matching toolkit MatchZoo[4], which integrate several text matching models.

[4] `https://github.com/faneshion/MatchZoo`

Table 4. Comparison between our proposed model and the baselines on the test set.

Models	SVM	LR	RF	Sm-CNN	Sm-LSTM	ARC-II	MP	MV-LSTM	BIMPM	**MTTM**
Acc(%)	61.59	63.57	60.99	65.68	66.17	67.23	66.84	66.52	67.48	**70.75**
F1(%)	71.18	74.41	70.57	75.94	76.00	76.20	74.68	77.34	77.40	**80.73**

Table 5. The correlation analysis between human scoring and our model's scoring in different metrics. All the correlation is significant with $p < 0.05$ (**Info** denotes informativeness, **Cons** denotes consistency, and **Nove** denotes novelty).

Correlation	**Info**	**Cons**	**Nove**	**Total**
Spearman	**0.740**	0.574	0.610	0.689
Pearson	**0.745**	0.544	0.608	0.704

- **MatchPyramid [11] (MP):** MatchPyramid transfers the traditional sentence matching task to an image recognition task.
- **MV-LSTM [13]:** The MV-LSTM model matches two sentences with multiple positional sentence representations.
- **BIMPM [14]:** BIMPM is a popular model to predict a label with matching two sentences. This model can match two texts from multi-perspectives. We implement this model according to the paper and related code.

4.3 Results

As is shown in Table 4 (All the results have passed the significance test), our proposed MTTM model achieves the best performance in the main evaluation metrics. MTTM can outperform both traditional machine learning methods and neural network methods. The BIMPM model has the best performance among all the baselines, and our MTTM model achieves improvements of 3.27% accuracy and 3.33% F_1 score over the BIMPM model. Compare to the existing text matching models, which usually focus on the matching between two kinds of texts, the MTTM model pays more attention to the difference of target texts and match the source text with each target text respectively. The experiment result shows that this multi-target text matching mechanism can learn better representation and improve the performance of classification.

4.4 Human Evaluation

In this paper, the popularity of comments is measured in three metrics: informativeness, consistence and novelty. Here come **two important questions**: can these metrics measure the comment popularity of comment well? Moreover, does our model realize the

Table 6. Ablation Study. Performance on the test set when removing different parts of text.

Models	**Acc(%)**	**F1(%)**
Full Model	70.75	80.73
w/o title	70.10($\downarrow$ 0.65)	79.79($\downarrow$ 0.94)
w/o abstract	69.82($\downarrow$ 0.93)	79.07($\downarrow$ 1.66)
w/o surrounding comments	69.16($\downarrow$ 1.59)	79.56($\downarrow$ 1.17)

Table 7. Performance on the test set with different number of surrounding comments.

Num	0	1	3	5
Acc(%)	69.16	68.22	69.53	**70.75**
F1(%)	79.56	78.59	79.18	**80.73**

measurement of the metrics successfully? Since these metrics are subjective, we use human evaluation and statistical analysis to analyze two questions. We randomly select 120 examples from the test set, and we assign three annotators(recruit from undergraduate of school) to evaluate the comments independently. Each comment is evaluated with a 5-point Likert-scale in three metrics: the informativeness of comment itself, the consistency between the comment and the news, and the novelty of comments compared with the surrounding comments. We average three annotators' scores for each metric to obtain the human scores. The results are scaled to [0,10].

To answer the **first question**, we analyze the relationship between the human scores and the popularity label of a comment. We conduct the independent sample $t-$test for annotators' score based on comment's popularity label. The results show that there are significant differences ($p < 0.05$) of the mean value of three human scores between popular comment class and common comment class. It concludes that the metrics we use in this work can measure the comment popularity well.

To analyze the **second question**, we obtain our model's scores on three metrics by mask different part of our model(set related matrix parameters to zero when predicting). We scale the output probabilities to [0, 10] so that it is comparable to the human scores. We conduct the correlation analyze between our model's scores and the human scores. We calculate the Pearson correlation coefficient and Spearman correlation coefficient for all the three scores as well as the total score. The result is shown in Table 5. We find that all these scores are significantly correlated ($p < 0.05$) between human and model's results. It concludes that our model realizes the measurement of three metrics successfully. Besides, among these three scores, the correlation coefficient of informativeness is highest, which indicates that the informativeness is more important in our model.

4.5 Impact of Different Parts of Text

Here we explore the impact of model inputs to its performance by removing different parts of the text. The result is shown in Table 6. As is shown in Table 6, the performance of the model shows different degrees of decline when we remove different context text. This shows that each input context is helpful for the classification and there are differences in the contribution of different context to the performance of the model. There is the smallest decline in the model performance when removing the title of news. It is reasonable because the news title tends to contain limited information.

4.6 Impact of the Surrounding Comment

In order to assess the novelty of comment, we also use the surrounding comments about this news as input text. Here the impact of the number of the surrounding comments on the model performance is further analyzed, and the related experiment result is shown in

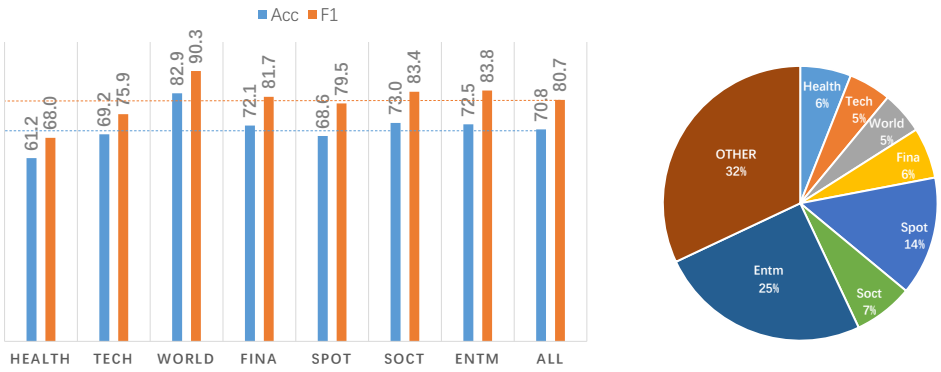

Fig. 2. Results of different types of news. The types from left to right are: health, technology, world, finance, sports, society, entertainment and average result.Different colors represents different metrics.Horizontal line represents the average level.

Table 7. According to Table 7, we find that with the increase in the number of surrounding comments, the model performs better, which shows that the surrounding comments are of great help for classification. The proposed model can refer to the surrounding comments for analyzing the novelty of a given comment. The larger the number of surrounding comments, the more input information can be enriched, leading to a more accurate assessment of novelty. However, we find that when only one surrounding comment is used, the performance of the model turns worse compared to using no surrounding comment. The reason is that the model suffers a significant variance in the case where there is only one comment, making the novelty score inaccurately evaluated.

4.7 Error Analysis

We find that there are significant differences in the performance of the model on different type of news. In order to explore the impact of the news type, we select seven different news types in our test set, and each type has at least several hundred samples. The performance of the model on these seven different types of news is shown in Figure 2. According to Figure 2, we find that the performance of the model on world news is better than average (accuracy 82.9% vs 70.8%, F_1 score 90.3% vs 80.7%). However, the model performance on the health news is worse than average (accuracy 61.2% vs 70.8%, F_1 score 68.0% vs 80.7%).

To analyze this phenomenon, we first count the number of news in each type in our training set. The result is shown in Figure 2. Moreover, we can see that the number of world news is close to health news. At the same time, the number of entertainment news is much larger than finance news, but they have a similar result in the test set. So we can conclude that the number of examples in the train set has little influence. Then why the results can be so different in world news and health news? We think it can be explained that world news contains less professional knowledge. So it is easy to arouse the user's

resonance to give reasonable feedback. At the same time, less professional knowledge makes it easy to capture the relevant semantic features, leading that the proposed model can learn an effective pattern to perform classification. However, there is a large amount of expertise in health news, leading to sparse data. Therefore, it is difficult for the model to learn a unified pattern for classification, resulting in poor performance.

5 Related Work

There have been some studies about news comments. [8] try to extract opinion target from news comments. Their method uses global information in news articles and contextual information in adjacent sentences of comments. [10] try to identify ''good'' online conversations. They build the Yahoo News comment threads Dataset and try to find Engaging, Respectful, and Informative Conversations. This dataset handles a thread of comment as a whole. [16] used a Graph-Structured LSTM to model the Reddit comment thread structure. However, we focus on the direct news comments which users read first and concern most. [6] also proposes a model to classifier the comments, and they focus on constructive comments. The dataset they use is rather small and lacks reliable annotation.
Siamese framework [2] is a classical method to deal with the Natural Language Sentence Matching(NLSM) task. However, the mutual information between the two sentences is lost in Siamese framework. [1] proposed Matching-Aggregation framework to overcome this problem. [4] proposed ARC-II model, which connects the n-gram of the two sentences and builds a 2D matrix first and then conduct matching. [11] proposed Match-Pyramid model, which transfers the text matching to image recognition by calculating the similarity matrix first. [15] find that attention architecture is helpful for the matching result. [14] propose BIMPM model, and they match the two sentences in two directions and multi-views on each hidden state of Bi-LSTM. [13] proposed the MV-LSTM model matches two sentences with multiple positional sentence representations.

6 Conclusions

In this work, we propose the task of popular comment prediction and construct a large-scale annotated dataset. We analyze the dataset and find the popularity of comments can be measured in three aspects: informativeness, consistency, and novelty. In order to measure three aspects above automatically, we propose a Multi-Target Text Matching model. Experimental results show that our model's scoring is highly correlated with human scoring in three aspects. Besides, our model outperforms various baselines by a large margin.

References

1. Bian, W.; Li, S.; Yang, Z.; Chen, G.; and Lin, Z. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 1987-1990.

2. Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Sackinger, E.; and Shah, R. 1993. Signature ‘ verification using A ‘‘siamese‘‘ time delay neural network. IJPRAI 7(4):669-688.
3. Graves, A.; Mohamed, A.; and Hinton, G. E. 2013. Speech recognition with deep recurrent neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26- 31, 2013, 6645-6649.
4. Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. Annual Conference on Neural Information Processing Systems 2014, 2042-2050.
5. Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980.
6. Kolhatkar, V., and Taboada, M. 2017. Using new york times picks to identify constructive comments. In Proceedings of the 2017 Workshop: Natural Language Processing meets Journalism, NLPmJ@EMNLP, Copenhagen, Denmark, September 7, 2017, 100-105.
7. Kolhatkar, V.; Wu, H.; Cavasso, L.; Francis, E.; Shukla, K.; and Taboada, M. 2018. The sfu opinion and comments corpus: A corpus for the analysis of online news comments.
8. Ma, T., and Wan, X. 2010. Opinion target extraction in chinese news comments. In COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 782-790.
9. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. 3111-3119.
10. Napoles, C.; Tetreault, J. R.; Pappu, A.; Rosato, E.; and Provenzale, B. 2017. Finding good conversations online: The yahoo news annotated comments corpus. In Proceedings of the 11th Linguistic Annotation Workshop, 2017, 13-23.
11. Pang, L.; Lan, Y.; Guo, J.; Xu, J.; Wan, S.; and Cheng, X. 2016. Text matching as image recognition. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2793-2799.
12. Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1):1929-1958.
13. Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X. (2016, March). A deep architecture for semantic matching with multiple positional sentence representations. In Thirtieth AAAI Conference on Artificial Intelligence.
14. Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral multiperspective matching for natural language sentences. In IJCAI 2017, 4144-4150.
15. Yang, L.; Ai, Q.; Guo, J.; and Croft, W. B. 2016. anmm: Ranking short answer texts with attention-based neural matching model. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 287-296. ACM.
16. Zayats, V., Ostendorf, M. (2018). Conversation modeling on Reddit using a graph-structured LSTM. Transactions of the Association of Computational Linguistics, 6, 121-132.