BLCU_NLP at SemEval-2019 Task 8: A Contextual Knowledge-enhanced GPT Model for Fact Checking

Wanying Xie, Mengxi Que, Ruoyao Yang, Chunhua Liu, Dong Yu(✉)
Beijing Language and Culture University, Beijing, China
{xiewanying07, quemengxi, yangruoyao97, chunhualiu596}@gmail.com
yudong@blcu.edu.cn

Abstract

Since the resources of Community Question Answering are abundant and information sharing becomes universal, it will be increasingly difficult to find factual information for questioners in massive messages. SemEval 2019 task 8 is focusing on these issues. We participate in the task and use Generative Pre-trained Transformer (OpenAI GPT) as our system. Our innovations are data extension, feature extraction, and input transformation. For contextual knowledge enhancement, we extend the training set of subtask A, use several features to improve the results of our system and adapt the input formats to be more suitable for this task. We demonstrate the effectiveness of our approaches, which achieves 81.95% of subtask A and 61.08% of subtask B in accuracy on the SemEval 2019 task 8.

1 Introduction

With the development of Community Question Answering (cQA) forums, massive information is being shared. However, not all information is factual, which makes finding an appropriate answer to satisfy the information needs of questioners more difficult. Previous work which concentrated on these problems (Nakov et al., 2017) reranked the questions based on their relevance with the original question. Šaina et al. (2017) treated the similarity ranking task as a binary classification problem. We study these issues in SemEval-2019 Task 8 (Mihaylova et al., 2019) by using the contextual Knowledge-enhanced GPT (Radford et al., 2018), which use Transformer (Vaswani et al., 2017) as model architecture. The contextual knowledge enhancement includes data extension, feature extraction, and input transformation.

The task includes two subtasks and they are both three classification problems. In subtask A, we need to find out whether a question seeks a factual answer, an opinion or just want to socialize with others. We classify the answers for questions that look for factual information in subtask A into three classes in subtask B: true, false or nonfactual. In this paper, we study both subtasks and use a similar system to solve them.

Several challenges exist when doing this task. The size of datasets for both subtasks is small. The data contains a number of complex long text, which makes extracting key information more difficult. The input format of GPT changes with different objectives of tasks, so it requires some modifications to fit specific tasks.

We apply three points to solve these problems. We extend the training set of subtask A from two other datasets: DailyDialog (Li et al., 2017) and SQuAD2.0 (Rajpurkar et al., 2018). We use two methods to guarantee the quality of expanded datasets. Firstly we use the Levenshtein Distance to screen similar data, and then we use the prediction of the model to further screen the results of the previous step. Goyal (2017) and Xie et al. (2017) used various features. Le et al. (2017) used keywords to solve the previous similar problem. We follow their work in feature extraction. Working on subtask A, we also use characteristic words as features to improve the system. Input transformation for classification task is $Start + Text + Extract$, including randomly initialized start and end tokens. We concatenate the text and features token sequences with a delimiter token.

The remainder of this paper is organized as follows. Section 2 contains a description of our system. The experiments and analysis of the results are introduced in section 3. We describe the conclusions in section 4.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1132–1137
Minneapolis, Minnesota, USA, June 6–7, 2019.

2 System Description

As Figure 1 shows, our system is composed of the following components: data extension, feature extraction, input transformation, and model.

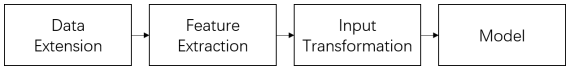

Figure 1: Flowchart of the system

Since the data provided by the task organizers are insufficient, our model does not get a high accuracy on such a small amount of data. We apply the data extension(Section 2.1) to address this problem. The extended datasets are DailyDialog and SQuAD2.0. We use two Levenshtein Distance and model prediction to ensure the expanded data similar enough to original data.

Since the data is composed of long text with complex information, it is difficult for our model to extract key information. We use feature extraction(Section 2.2) to solve this problem, which is able to bring high discrimination between data categories. We add two kinds of features to the input: original features directly extracted from the training data and observed features is summarized by us. After feature extraction, the key information gets enhanced and our model has easier access to significant information of data.

Different types of tasks correspond to different input transformation of GPT. We change the input transformation(Section 2.3) to fit different tasks. If a task is about classification, the input format is supposed to be $Start + Text + Extract$. We need to adjust the input of our model to adapt to the specific task, also need to add features in the data. We use special character $Delim$ to connect different features and the main text, then we use the connected formats as our input.

2.1 Data Extension

This section briefly introduces the datasets of subtask A, subtask B, DailyDialog and SQuAD2.0. Then we introduce data extension from DailyDialog and SQuAD2.0. Figure 2 shows the total process of extending data for subtask A. We use two approaches to ensure the data that we expanded from other datasets similar enough with original data. The two approaches are Levenshtein Distance screening and model prediction screening.

Data Overview The datasets we used for subtask A and subtask B are provided by the task organizers. We extend the training set of subtask A from two other datasets: DailyDialog and SQuAD2.0.

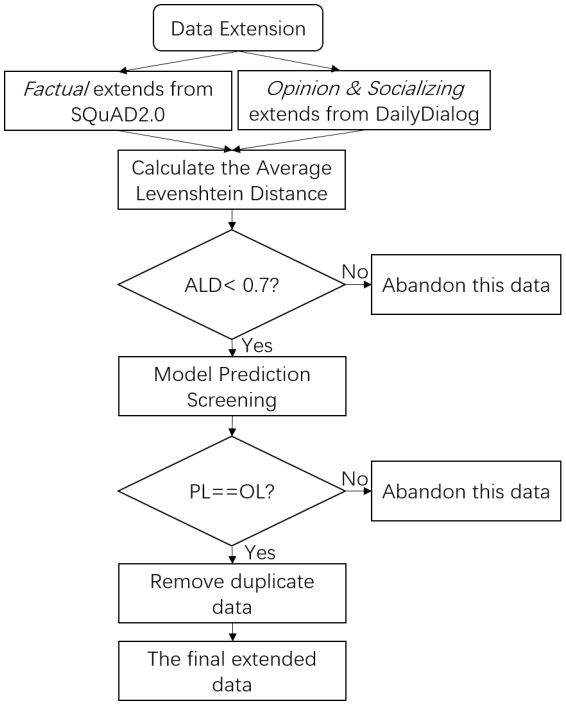

Figure 2: Flowchart of data extension. ALD represents average Levenshtein distance. PL stands for perdicted label and OL stands for original label. Since *Opinion* and *Socializing* are both extended in Daily Dialog, so we remove duplication in the alternative extended datasets of *Opinion*.

- In subtask A, there are 1118 threads in the original dataset. Each thread consists of RelQBody, RelQSubjcet, RelQCategory, RelQDate, RelQId, RelQUser Id and RelQUsername. RelQBody is a complete question description but its text is too long. RelQSubject is short but lacks feature information. RelQCategory is mentioned to show the question's category and it is useful in classifying the questions. Our emphasis is subtask A, which is working on deciding the questions' classification label *Factual*, *Opinion* and *Socializing*. Specifically, the size of each class is 311, 563 and 244. The *Opinion* class accounts for more than half of the total data.

- The dataset of Subtask B is similar to subtask A. There are 495 answers in the original dataset. One question corresponds to one or several answers. The purpose of subtask B

is to divide the answers into three categories: *True*, *False* and *Nonfactual*.

- DailyDialog is a multi-turn dialog dataset, which includes questions and answers, and the data size is 11318. The topic of DailyDialog is about daily life, so we consider that we expand *Opinion* and *Socializing* label data from it.

- SQuAD2.0 is a reading comprehension dataset, which intends to answer a question according to the context. Absolutely, questions in this dataset all ask for factual information since the answers can be found from the context. We decided to extend the *Factual* label data from the questions of this dataset.

Levenshtein Distance Screening Levenshtein distance is also known as Editing distance, which refers to the smallest number of editing operations required to change one string into another. The editing operation consists of three choices: replacing one character with another, inserting one character, and deleting one character. When comparing the two sentences, they will be more similar if the Levenshtein distance is smaller.

Levenshtein distance is used to calculate the similarity between two sentences. Since the length of sentences in the dataset is uncertain, the Levenshtein distance is an integer of indeterminate size. We divide the Levenshtein distance by a larger length of two sentences. Ultimately, what we get is not an unlimited integer, but a decimal between 0 and 1. In this paper, it is called average Levenshtein distance, as shown in Equation(1), where ALD(s1,s2) represents the average Levenshtein distance of sentence 1 and sentence 2.

$$ALD(s1, s2) = \frac{LevenshteinDistance}{max(len(s1), len(s2))} \quad (1)$$

We regard the question data of the cQA(community QA) forum as the original data and divide it into three categories according to the different labels of the questions. When traversing instance in *Opinion* and *Socializing*, the ALD between the original data and the question of DailyDialog data is calculated. When traversing the data of *Factual*, the ALD between the original data and the question of SQuAD 2.0 data is calculated.

We set a threshold of **0.7**. At this threshold, we are able to get more data which is guaranteed to be sufficiently similar. After calculating the ALD, if the value is less than the threshold, it means the two sentences are sufficiently similar. Then we use this data as alternative extended data. Finally, we get three alternative extended datasets with their original label. Since *Opinion* and *Socializing* are both extended from DailyDialog, it will be some plication in the two extended datasets. The original *Socializing* data is less than *Opinion*, so we remove duplication in the extended datasets of *Opinion* in order to get roughly the same amount of data. It means if a data appears in the alternative extended datasets of *Socializing*, then remove this data in the alternative extended datasets of *Opinion*.

Model Prediction Screening The question dataset of the cQA forum is used as the training set to train the GPT model, and the candidate extended dataset is used as the test set to predict. If the predicted label is consistent with the original label of the test set, then this data is considered to be correct prediction data. As one of the extended datasets, if the predicted label is inconsistent with the original label, which means that it is wrong, it is considered that this data is not helpful to the model, so we discard this data.

	original	extended	sum
factual	311	434	745
opinion	563	308	871
socializing	244	598	842

Table 1: Class distribution of subtask A

After screening by Levenshtein distance and model prediction, we finally get 434, 308 and 598 for *Factual*, *Opinion* and *Socializing* to expand. After expansion, the number of three classified data is 745, 871 and 842, respectively. Table 1 shows the total data category distribution of the subtask A.

2.2 Feature Extraction

We introduce two kinds of features and explain how we apply these features to the GPT model in detail.

Features Acquisition Subtask A includes two kinds of features. One we call *Original Features* is given directly in the dataset. The other we call *Observed Features* is obtained from the data observation. For subtask B, we just use the features

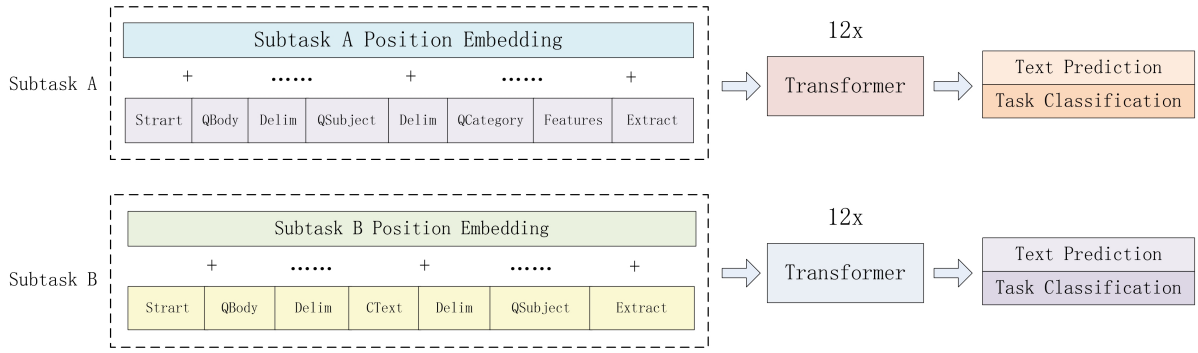

Figure 3: Input formats of subtasks. The input formats consist of text embedding and position embedding. Subtasks A and B input to the model separately. In the model, it is a 12-layer decoder-only transformer with masked self-attention heads. Text prediction and task classification are both the fine-tuning objective of the GPT model.(Radford et al., 2018)

extracted directly in the dataset. We present two kinds of features as follows.

Original Features In the dataset of subtask A, each thread consists of RelQBody, RelQSubjcet, RelQCategory, RelQDate, RelQId, RelQUser Id and RelQUsername. The main text is RelQbody, and we consider other information as features. Through our screening, it is a suitable method to regard RelQSubject and RelQCategory as original features. For extended data, they are no original features.

For subtask B, we choose RelQbody and RelQSubject as the original features.

Observed Features The second kind of feature is obtained from data observation in subtask A. In *Factual* data, there are a lot of questions about Visas for couples, working, pets and animals, opportunities, etc. In *Opinion* data, questions that ask for advice are more common. In *Socializing* data, the questions are more colloquial, so it may include a word like *qler*. We chose the observed characteristic words as features. There are several examples for each class:

- *Factual*: *visit, license, husband, wife, embassy, sponsor*
- *Opinion*: *advice, school, suggestion, advise*
- *Socializing*: *ql, qler, weekend, love, going, today*

2.3 Model

Input Transformation Input sequence contains three special characters $Start$, $Extract$ and $Delim$, representing the start, end, and delimiter token respectively. We treat subtask A and B as question classification tasks. Their input formats are as follows.

- Subtask A: The most useful information in the data is the complete problem description RelQBody. We choose it as the main text and RelQSubject and RelQCategory as the original features. We employ the main text and two original features as the input. The observed features are also added to the input. The final input representation is $Start + RelQBody + Delim + RelQSubject + Delim + RelQCategory + Features + Extract$.
- Subtask B: We use RelCText as the main text, and use RelQBody and RelQSubject as original features. These features constitute the model input. We do not employ observed features in subtask B. So the final input format is $Start + RelQBody + Delim + RelCText + Delim + RelQSubject + Extract$

Model Description GPT is a language model, pre-trained on BooksCorpus (Zhu et al., 2015). There are 12-layer *Transformer* blocks. When optimizing, the training loss is the sum of text prediction loss and classification loss. Different tasks correspond to different input formats when using the GPT model for fine-tuning. Subtasks A and B input separately to the model. Figure 3 shows the input formats in detail.

3 Experiments

We present the experiments we conduct on our system and make a detailed analysis. We compare the performance between GPT and other models

in subtask B. Follow Radford et al. (2018), we use the default model configuration for our model.

3.1 Results

We evaluate the systems on the development set and use accuracy as the main evaluation criteria. We use the organizer's score on the practice leaderboard of CodaLab as the baseline. Table 2 shows our performances on subtask A in detail. Original F and Observed F represent original features and observed features respectively. DE means Data Extension which we mentioned in section 2.1. Our best system achieves 81.59% in the development set.

	Acc	F1	AvgRec
GPT	0.7768	0.6392	0.6392
Above+Original F	0.7964	0.6738	0.6721
Above+Observed F	0.7992	0.6795	0.6771
Above+DE	**0.8159**	0.6959	0.6859

Table 2: Development result of subtask A. Acc means accuracy. Above means that a new change is added to the system which mentioned in previous row.

Table 3 shows our performances on subtask B. **F** represent features in the original dataset. We get 69.05% in the development set.

	Acc	F1	AvgRec	MAP
GPT	0.6369	0.4207	0.4312	0.7889
GPT+F	**0.6905**	0.4848	0.4789	0.7500

Table 3: Development result of subtask B. Acc means accuracy.

Using our best system we evaluate in the test set. As Table 4 shows, the score of our official submission is 81.95% in subtask A, which ranks sixth in all participants. The baseline of the test set is 45.0% in accuracy which lower than all the participants' score. In subtask B, we achieve 61.08% in the test set, which ranks seventh in all participants. The baseline of subtask B is 83.0%.

	Subtask A	Subtask B
Baseline	0.450	**0.830**
Our System	**0.8195**	0.6108

Table 4: Official submissions results on the test set for our system and the organizers's baselines. The metric is accuracy.

3.2 Analysis

Subtask A Adding original features proves to be useful to GPT, which increases by 2% than single GPT in accuracy. Observed features are not as useful as original features are. They only improve the result slightly. Data extension is also helpful, which improves the score by 1.67%.

Subtask B Original features are helpful for GPT and increase accuracy by 5.36%. It is a great improvement. The possible explanation might be that original features provide key information to the classification task in subtask B. However, their performance on the test set is not satisfactory, which only achieves 61.08%.

We implement the ESIM model (Chen et al., 2016) in subtask B, which applies bidirectional Long Short Term Memory (Hochreiter and Schmidhuber, 1997) and an alignment mechanism, achieving 63.69% in accuracy of the development set. Furthermore, we concatenate glove embeddings with contextual embeddings produced by ELMo (Peters et al., 2018) as features, improving accuracy of the development set by 2% in subtask B. Both results in subtask B are less than the best result of GPT, which is 69.05%. So we use GPT as our official system in subtask B.

4 Conclusions

We use the GPT model to participate in SemEval 2019 task 8. The goal of this task is question classification and answer classification. We demonstrate that large gains on fact checking can be realized by data extension, feature extraction, and input formats transformation. Our official submission achieves accuracy 81.95% of subtask A and 61.08% of subtask B, which ranks us 6th and 7th in the competition. What's more, features and data expansion are both helpful to the system.

For future work, we think data extension may be useful in subtask B since it performs well in subtask A. Furthermore, we would like to use external information in this task.

Acknowledgments

This work is funded by Beijing Advanced Innovation for Language Resources of BLCU TYR17001J the Fundamental Research Funds for the Central Universities in BLCU (No.17PT05) and the BLCU Academic Talents Support Program for the Young and Middle-Aged.

References

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Naman Goyal. 2017. Learningtoquestion at semeval 2017 task 3: Ranking similar questions by learning to rank using rich features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 310–314.

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

Qi Le, Zhang Yu, and Ting Liu. 2017. Scir-qa at semeval-2017 task 3: Cnn model based on similar and dissimilar information between keywords for question similarity. In *International Workshop on Semantic Evaluation*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Tsvetomila Mihaylova, Georgi Karadzhov, Atanasova Pepa, Ramy Baly, Mitra Mohtarami, and Preslav Nakov. 2019. SemEval-2019 task 8: Fact checking in community question answering forums. In *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '19, Minneapolis, MN, USA.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48.

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, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Filip Šaina, Toni Kukurin, Lukrecija Puljić, Mladen Karan, and Jan Šnajder. 2017. Takelab-qa at semeval-2017 task 3: Classification experiments for answer retrieval in community qa. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 339–343.

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.

Yufei Xie, Maoquan Wang, Jing Ma, Jian Jiang, and Zhao Lu. 2017. Eica team at semeval-2017 task 3: Semantic and metadata-based features for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 292–298.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.