X-PuDu at SemEval-2022 Task 7: A Replaced Token Detection Task Pre-trained Model with Pattern-aware Ensembling for Identifying Plausible Clarifications

Junyuan Shang[1], **Shuohuan Wang**[1], **Yu Sun**[1],
Yanjun Yu[2], **Yue Zhou**[2], **Li Xiang**[2] and **Guixiu Yang**[2]
[1] Baidu Inc., China
[2] Shanghai Pudong Development Bank, China
{shangjunyuan, wangshuohuan, sunyu02}@baidu.com
{yuyj6, zhouy93, xiangl3, yanggx1}@spdb.com.cn

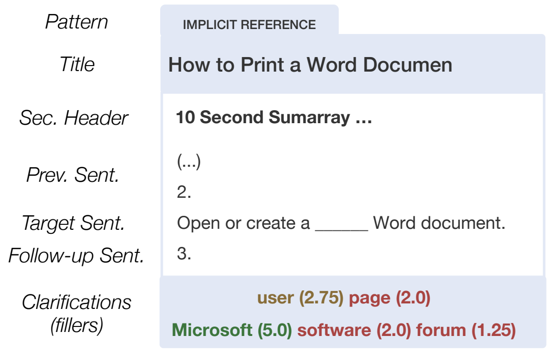
Abstract

This paper describes our winning system on SemEval 2022 Task 7: *Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts*. A replaced token detection pre-trained model is utilized with minorly different task-specific heads for SubTask-A: *Multi-class Classification* and SubTask-B: *Ranking*. Incorporating a pattern-aware ensemble method, our system achieves a 68.90% accuracy score and 0.8070 spearman's rank correlation score surpassing the 2nd place with a large margin by 2.7 and 2.2 percent points for SubTask-A and SubTask-B, respectively. Our approach is simple and easy to implement, and we conducted ablation studies and qualitative and quantitative analyses for the working strategies used in our system.

1 Introduction

The Internet's ever-increasing size has made it easy to find instructional texts such as articles in wikiHow[1], on almost any topic or activity. Regular revisions of these how-to manuals are necessary to ensure that instructions communicate the procedures required to attain a certain goal precisely. This shared task is introduced by Roth et al. (2022), whose intention is to find ways to improve instructional texts, evaluate to what extent current NLP systems are able to handle implicit, ambiguous, and underspecified language, and go beyond the surface form of a text and take multiple plausible interpretations into account. Thus, the proposed NLP systems should be capable of distinguishing between plausible and implausible clarifications of an instruction shown in Figure. 1.

The shared task consists of two subtasks:



Pattern	IMPLICIT REFERENCE
Title	How to Print a Word Documen
Sec. Header	10 Second Sumarray ...
Prev. Sent.	(...) 2.
Target Sent.	Open or create a ______ Word document.
Follow-up Sent.	3.
Clarifications (fillers)	user (2.75) page (2.0) Microsoft (5.0) software (2.0) forum (1.25)

Figure 1: An example randomly chosen from the dev set. Each sample is associated with five clarifications labeled (PLAUSIBLE, NEUTRAL or IMPLAUSIBLE) and scored on a scale from 1.0 to 5.0.

- **SubTask-A: Multi-Class Classification**. The goal is to predict a class label (IMPLAUSIBLE, NEUTRAL, PLAUSIBLE) given the clarification[2] and its context.
- **SubTask-B: Ranking**. The goal is to predict the plausibility score on a scale from 1 to 5 given the clarification and its context.

In this paper, we describe our winning system for both subtasks. We built our system based on a replaced token detection (RTD) task pre-training model. The idea is that the replaced token detection task is similar to this shared task which focuses on distinguishing semantically similar words/phrases. To close the gap the model trained between the pre-training phase and fine-tuning phase, we reused the pre-trained language modeling head during fine-tuning on Task 7. Then, two-layers MLPs are applied on the mean-pooled hidden states of a clarification (filler) given the context. For subtask A, we utilized the cross-entropy loss for the multi-class classification. For subtask B, a sigmoid function was used to impose restrictions on the output of the system on a scale from 1 to 5. Finally, we

[1] https://www.wikihow.com/Main-Page

[2] A clarification is a word/phrase that was inserted to specify information in the instruction.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1078 - 1083
July 14-15, 2022 ©2022 Association for Computational Linguistics

trained multiple models and aggregated the predictions with a pattern-aware ensemble strategy. Our system achieved the best overall performance in the shared task with a 68.9% accuracy score (subtask A) and 0.807 Spearman's rank correlation score (subtask B). The outcomes are promising for improving the clarification of instructional texts.

2 Background

Pre-trained models (Devlin et al., 2018; Liu et al., 2019; Sun et al., 2019; Clark et al., 2020) have achieved state-of-the-art results in various Natural Language Processing (NLP) tasks. Recent works (Raffel et al., 2019; Brown et al., 2020; Sun et al., 2021) have shown that more generalization ability and superior performance can be achieved by pre-training models with billion or trillion parameters. Thus, we pursued the competitive pre-trained models such as DeBERTa (He et al., 2020) and large-scale pre-trained models ERNIE (Sun et al., 2021) whose effectiveness has been validated in the standard GLUE (Wang et al., 2018) and SuperGLUE benchmark (Wang et al., 2019).

However, in our initial experiment, we found that the aforementioned models, though have promising results on sentence-level or paragraph-level tasks, failed to distinguish the word/phrase-level semantically similar clarifications (fillers) in a given context. We believe the failure is due to the way these models were trained using a masked language modeling (MLM) task in the pre-training phase. MLM aims to map tokens with similar semantics to the embedding space that are close to each other instead of distinguishing them.

Based on the above finding, we believe what we need is a discriminator (Clark et al., 2020; He et al., 2021) pre-trained via a replaced token detection (RTD) task which is more aligned with this shared task. In RTD, the discriminator needs to determine if a corresponding token is either an original token or a token replaced by the generator. Formally, the loss function for the discriminator is as follows:

$$\mathcal{L}_{RTD} = -\sum_{i} \log p \left(\mathbb{1} \left(\tilde{x}_i = x_i \right) \mid \tilde{\mathbf{X}}, i \right) \quad (1)$$

where $\tilde{\mathbf{X}}$ is the input sequence constructed by replacing masked tokens with plausible tokens sampled from a generator, and the indicator function $\mathbb{1}(\cdot)$ distinguishes whether the plausible tokens are generated or the original ones.

3 Method

In this section, we will describe the strategies we used in our system in detail. In Section. 3.1, the system is presented on how we formalize the data as input, basic modules, and task-specific design for each subtask. Then, we describe the optimization object for each subtask (see Section. 3.2). Finally, we introduce a pattern-aware ensemble strategy to further boost the performance beyond a normal ensembled model in Section. 3.3.

3.1 System Description

As illustrated in Figure. 2, the framework for both tasks is nearly the same which consists of 4 parts, namely the input, a basic model, a pre-trained head, and a task-specific head.

The input sequence. Each sample is constructed by joining the *Pattern*, *Title*, *Section Header*, *Previous Sentence*, *Target Sentence* and *Follow-up Sentence* in order demonstrated in Figure. 1. Each candidate phase is filled in in its original position in the *Target Sentence*. When modeling the filled target sentence independently, the training set will be 5 times larger than the original since each target phrase has five candidates.

Basic Model. The pre-trained transformer is our starting point. The basic model takes as input the sequence $\tilde{x}$ and outputs the contextual representation of each token as follows:

$$\mathbf{H}_b = \text{Transformer}(\tilde{x}) \quad (2)$$

where $\mathbf{H}_b \in \mathbb{R}^{n \times d}$ with n tokens and d dimension.

Pre-trained Head. During the pre-training phase, a language modeling head is appended for a language modeling task. The head is usually discarded in the fine-tuning phase. However, in our experiment, we found better performance can be achieved when reusing the pre-trained head. Formally, the language modeling head takes as the input $\mathbf{H}_b$ and output representations for task-specific head as follows:

$$\mathbf{H}_p = \text{LN}(\text{Act}(\mathbf{H}_b \mathbf{W}_1 + \mathbf{b}_1)) \quad (3)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d}, \mathbf{b}_1 \in \mathbb{R}^d$ is the weight and bias, $\text{Act}(\cdot)$ and $\text{LN}(\cdot)$ are the activation function and the layernorm layer (Ba et al., 2016) respectively.

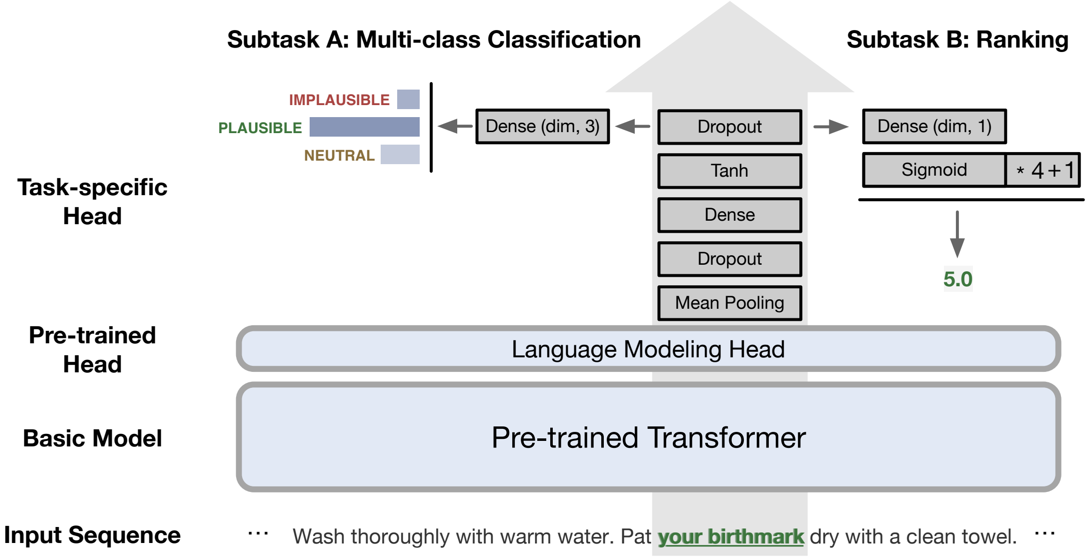


Figure 2: The illustration of our system.

Task-specfic Head. As there are several tokens after tokenizing the target phrase, we apply a mean pooing layer for the hidden states of the target phrase denoted as $\mathbf{H}_{p,i:j}$ as follows:

$$\mathbf{h}_t = \frac{\sum_i^j \mathbf{H}_{p,i}}{j-i} \tag{4}$$

where $\mathbf{h}_t \in \mathbb{R}^{1\times d}$ is the mean embedding of the target phase, i, j are the start and end token index of the tokenized target phrase respectively. Then, we sequentially appended a dropout layer, a dense layer with a Tanh activation function and a dropout layer for $\mathbf{h}_t$ as follows:

$$\tilde{\mathbf{h}}_t = \text{Dropout}(\text{Tanh}(\text{Dropout}(\mathbf{h}_t)\mathbf{W}_2 + \mathbf{b}_2)) \tag{5}$$

where $\tilde{\mathbf{h}}_t \in \mathbb{R}^{1\times d}, \mathbf{W}_2 \in \mathbb{R}^{d\times d}, \mathbf{b}_2 \in \mathbb{R}^d$ are the enhanced embedding of the target phase, learnable weight and bias respectively. Finally, the $\tilde{\mathbf{h}}_t$ is transformed to fit the three-class classification task and regression task as follows:

$$\tilde{\mathbf{y}}_c = \text{Softmax}(\tilde{\mathbf{h}}_t \mathbf{W}_3 + \mathbf{b}_3) \tag{6}$$

$$\tilde{y}_r = \text{Sigmoid}(\tilde{\mathbf{h}}_t \mathbf{W}_4 + \mathbf{b}_4) * 4 + 1 \tag{7}$$

where $\tilde{\mathbf{y}}_c \in \mathbb{R}^{1\times 3}, \mathbf{W}_3 \in \mathbb{R}^{d\times 3}, \mathbf{b}_3 \in \mathbb{R}^3$ are the probabilty distribution, learnable weight and bias for subtask A, and $\tilde{y}_r \in \mathbb{R}^1, \mathbf{W}_3 \in \mathbb{R}^{d\times 1}, \mathbf{b}_3 \in \mathbb{R}^1$ are the regression score, learnable weight and bias for subtask B. The Sigmoid function restricts the range of output space between 0 to 1, then we shift the number by multiplying four and adding one. The above method successfully restrict the regression score within the golden score on a scale of 1 to 5.

3.2 Optimazation Object

For subtask A, we utilized the cross-entropy loss for multi-class classification as follows:

$$\mathcal{L}_{ce} = -\sum_i^N \log(\tilde{\mathbf{y}}_c^i[y_c^i]) \tag{8}$$

where N is the number of training samples, y_c^i is the golden label for i-th sample, $\tilde{\mathbf{y}}_c^i[y_c^i]$ means the predicted probability of the golden label.

For subtask B, we used the mean squared error loss for regression as follows:

$$\mathcal{L}_{reg} = \frac{1}{N}\sum_i^N (\tilde{y}_r^i - y_r^i)^2 \tag{9}$$

3.3 Pattern-aware Ensembling

Ensemble is the commonly used technique where multiple diverse models are trained to predict an outcome, then aggregates the prediction of each model resulting in the final prediction. In our experiment, we observed that the model fine-tuned with different hyper-parameters have different preference on the *Resolved Pattern*[3]. Thus, we aggregates the prediction of each model seperately based on the performance on a subset split by the given *Resolved Pattern* attribute.

[3] Descriptions of the resolved pattern can be found in https://competitions.codalab.org/competitions/35210#participate

Pattern\Dataset	Train	Validation	Test
ADDED COMPOUND	5000	625	625
FUSED HEAD	4995	625	625
IMPLICIT REFERENCE	4980	625	625
METONYMIC REFERENCE	5000	625	625
Total	19975	2500	2500

Table 1: Data statistics in train, validation, and test set on the different patterns.

Hyper-parameter	Model
Dropout	0.1
Warmup Ratio	0.1
Learning Rates	{5e-6, 7e-6, 9e-6, 1e-5}
Batch Size	{32, 48, 64}
Weight Decay	0.01
Epoches	5
Learning Rate Decay	Linear

Table 2: Hyper-parameters for fine-tuning on both subtasks.

4 Experiment

4.1 Data

We use the training, validation and test data provided for SemEval 2022 Task 7 without introducing extra data. The data statistic is summarized in Table. 1 where there is a balanced distribution among different patterns.

4.2 Experimental Setup

DeBERTa (He et al., 2020, 2021), XLMR (Conneau et al., 2019) and ERNIE (Sun et al., 2021) are used as the pre-trained language model. We fine-tune the models using the AdamW optimizer (Kingma and Ba, 2014) with the default hyper-parameter, and additional fine-tuning hyper-parameters are listed in Table. 2. Experiments are carried out using eight Nvidia A100 GPUs.

4.3 Evaluation Method

For subtask A, the evaluation metric is the accuracy score. The model must predict one of the following labels: {IMPLAUSIBLE, NEUTRAL, PLAUSIBLE}.

For subtask B, the submission will be scored using Spearman's rank correlation coefficient, which compares the predicted plausibility ranking over all test samples to the gold ranking.

Task	**SubTask-A**		**SubTask-B**	
	Dev	Test	Dev	Test
2nd Place	-	66.10	-	0.7850
Ensembled Model	71.08	66.50	0.8260	0.7950
+ Pattern-aware	**75.20**	**68.90**	**0.8441**	**0.8070**

Table 3: Performance of models on dev set and official test set.

#	**Models**	**SubTask-A**
	MLM-based Models	
1	XLMR-Large	61.14
2	ERNIE	61.73
	RTD-based Models	
3	DeBERTa-V3-Large	**67.25**
4	#3 without pre-trained head	65.96

Table 4: Ablation studies on SubTask A with respect to the accuracy score on the dev set. (We reported the mean results with at least three runs.)

4.4 Results

Our ensembled prediction on test set placed first in the competition, with a 68.9% accuracy score for subtask A and a 0.8070 Spearman's rank correlation coefficient for subtask B. As shown in Table. 3. Our system outperforms the second-place system by 2.8 and 2.2 percent points respectively. The organizers predict an upper bound of 77.1% accuracy score and 0.89 ranking correlation based on the manual annotations. As a result, there's still a lot of room for growth.

4.5 Ablation Studies

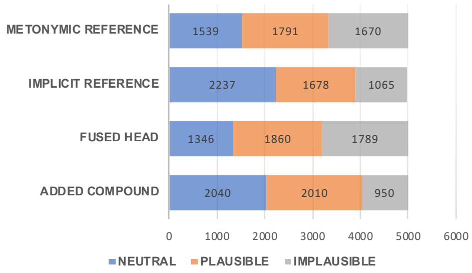


Figure 3: The label distribution of different pattern on training dataset.

The effectiveness of using a replaced token detection task pre-trained model and recovering the pre-train language modeling head in the task-specific

Pattern	IMPLICIT REFERENCE		METONYMIC REFERENCE		FUSED HEAD		ADDED COMPOUND	
Hyper-paramters\Task	SubTask-A	SubTask-B	SubTask-A	SubTask-B	SubTask-A	SubTask-B	SubTask-A	SubTask-B
LR:1e-5, BSZ:32	65.12	0.8321	67.36	**0.8427**	67.68	0.8400	64.64	**0.8272**
LR:9e-6, BSZ:32	64.96	**0.8340**	**69.60**	0.8408	**71.84**	**0.8418**	63.20	0.8251
LR:1e-5, BSZ:64	71.84	0.8286	69.28	0.8382	65.12	0.8347	68.96	0.8134
LR:9e-6, BSZ:64	**72.32**	0.8265	**69.60**	0.8424	64.96	0.8325	**69.28**	0.8142

Table 5: Performance of the DeBERTa-V3-Large with different fine-tuning hyperparameters on the dev set. A model can't win all the subtasks on a subset split by the given pattern attribute. (LR and BSZ are abbreviations for learning rate and batch size.)

head have been revealed in Table. 4. The hypothesis that utilizing a model pre-trained by a similar task aligned with SemEval-2022 Task 7 contributes a lot is supported by comparing #1,#2 and #3. The performance of the model improved even more after reusing the pre-trained LM head (#3 and #4). The assumption is that the hidden states from the pre-trained head contain more information learned during the pre-training phase for distinguishing semantically similar tokens.

The effectiveness of the pattern-aware ensembling has been shown in Table.3. On subtasks A and B, pattern-aware ensembling outperformed the standard ensemble technique by 2.4 and 1.2 percent points, respectively, compared to the standard ensemble method.

The model trained with different hyperparameters may perform better on one pattern but not on another, as seen in Table. 5. For example, on the `FUSED HEAD` pattern, the model (LR:9e-6, BSZ:32) has the highest accuracy score of 71.84% but the lowest accuracy scores of 64.96% and 63.20% on `ADDED COMPOUND` and `IMPLICIT REFERENCE` pattern, respectively. The model (LR:9e-6, BSZ:64), on the other hand, has the lowest score on `FUSED HEAD` pattern but the best result on `ADDED COMPOUND` and `IMPLICIT REFERENCE` pattern. By visualizing the label distribution in Figure. 3, we infer that the phenomenon is related to a distribution difference in which `FUSED HEAD` pattern contains the lowest number of the label *NEUTRAL*, and the label `PLAUSIBLE` dominates the `ADDED COMPOUND` and `IMPLICIT REFERENCE` patterns.

5 Conclusion

We built a system for identifying plausible clarifications of implicit and underspecified phrases in instructional texts which is useful for improving the clarification of instructional texts. The system leverages the strength of a replaced token detection pre-trained discriminator and therefore performs extremely well on this shared task with the same goal to distinguish semantically similar tokens. In particular, we proposed a pattern-aware ensembling strategy to aggregate multiple predictions separately based on the pattern when there is a label distribution difference among patterns. On SemEval-2022 Task 7, the system achieved the best performance in both subtasks.

In future work, it's promising to incorporate the replace token detection task in a large-scale pre-trained model with billion, or even trillion parameters.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Michael Roth, Talita Anthonio, and Anna Sauer. 2022. SemEval-2022 Task 7: Identifying plausible clarifications of implicit and underspecified phrases in instructional texts. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

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*.