More Semantically Focused Modeling for Semantic Text Matching

Anonymous ACL submission

Abstract

As is widely acknowledged, Pre-trained Language Models (PLMs) acquire the capability to encode deep sentence semantics through pre-training. Semantic Text Matching (STM) task has greatly benefited from this capacity. However, the extent to which PLMs can fully exploit semantic encoding, rather than merely relying on some superficial pattern recognition in this task, remains a matter for investigation. We argue that a model's ability to provide consistent judgments despite variations in phrasing indicates its reliance on semantic interpretation. Based on this perspective, we investigate the extent to which the model captures semantics and introduce a novel training architecture aimed at enhancing the semantic modeling capacity of PLMs in STM tasks. Our approach is validated through rigorous experimentation on four benchmark datasets: LCQMC, BQ, QQP, and MRPC, where we achieve state-of-the-art performance on three of them.

1 Introduction

Natural Language Understanding (NLU) is a subfield of natural language processing that focuses on machine reading comprehension. Natural language is merely a vehicle for humans to convey thoughts. At their core, they are nothing but symbols whose meanings are defined by humans. In other words, for humans, we create "language" for the purpose of "semantics". When individuals employ language to describe something or express an emotion, the resulting expressions may vary significantly among different people due to diverse habits, personal experiences, and other factors. Even the same individual may produce expressions of entirely different styles when in varying environments or states of mind. Nonetheless, people are invariably able to effortlessly comprehend the fundamental semantics underlying these diverse expressions. The question then arises: can deep neural network models achieve a similar level of understanding?

Recently, significant advancements have been made in NLU due to the use of PLMs, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Benefiting from pre-training on large-scale textual corpora, PLMs model the semantics of these linguistic symbols and acquire a substantial amount of prior knowledge. That is to say, for the model, it is a matter of inferring "semantics" from "language". STM, a sub-task within NLU that aims to determine the semantic similarity of two sentence-level texts, has also seen significant advancement with PLMs. However, the extent to which model performance on this task relies on deep semantic modeling, as opposed to the exploitation of shallow patterns, such as lexical similarity, remains an open question. We contend that the ability of a model to achieve a level of understanding comparable to humans, such that it renders consistent judgments regardless of variations in phrasing, is a critical manifestation of its true reliance on semantic interpretation for decision-making.

This paper investigates the semantic modeling ability of PLMs in the STM task and proposes a training architecture to enhance this ability. We experiment on four datasets, including LCQMC, BQ, QQP, and MRPC, and achieve state-of-the-art performance on three of them. Our code is available at the supplementary material.

2 Related work

Previous research has indicated that BERT models excessively rely on lexical overlap between text pairs when determining semantic similarity (Zhang et al., 2019; Yu and Ettinger, 2020; Wang et al., 2021, 2022). We posit that this implicitly reveals that models merely engage in superficial modeling of textual symbols, where texts that appear similar are presumed to have analogous meanings, without sufficiently modeling the semantics underlying these symbols.

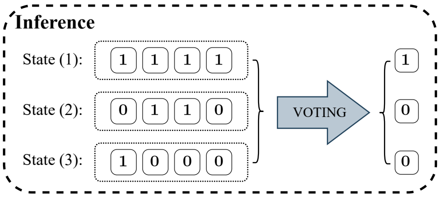

Figure 1: Examples of voting strategy.

Recently, there has been a rapid evolution in Large Language Models (LLMs). LLMs demonstrate remarkable capabilities in following instructions and generating smooth, natural-sounding language. While the issue of hallucinations in LLMs during complex tasks remains a significant unresolved challenge (Zhang et al., 2023), for relatively straightforward tasks such as rewriting sentences without altering their meaning, we can leverage their instruction-following generative capabilities to the fullest extent while maintaining hallucination issues at an acceptable level.

3 Methodology

For STM tasks, a sample X consists of a pair of input texts, (S_a, S_b), along with their corresponding label l. Firstly, we use a LLM to generate paraphrased sentences R_a and R_b corresponding to the text pair S_a and S_b. Subsequently, by combining them, we obtain a set $\hat{X}$ that encompasses four distinct representations of the sample X.

$$\begin{aligned} R_a, R_b &= \text{LLM}(T(S_a)), \text{LLM}(T(S_b)) \\ \hat{X} &= \{\, (S_a, S_b), (S_a, R_b), (R_a, S_b), (R_a, R_b) \,\}, \end{aligned} \tag{1}$$

where $T(*)$ is the template of instruction. Given that our instruction requires the LLM to rewrite texts while ensuring that the meaning remains unchanged, the semantics of R_a and R_b are consistent with S_a and S_b, albeit with differences in syntactic structure or word choice.

To investigate whether models can make consistent predictions for sentence pairs within the same $\hat{X}$, we initially train a BERT-based baseline model on the LCQMC dataset, which is trained on the original training set and subsequently performed inference on the original test set. The results are illustrated in the first row of Table 1.

Then, for each sample X in the test set, we apply Formula 1 to obtain the corresponding expanded $\hat{X}$, and predict the four text pairs within $\hat{X}$ using the baseline model. Consequently, for each sample, we obtain four prediction outcomes. We categorize these outcomes into three states as follows: (1) If all prediction results are in agreement, we consider that the model can understand the semantics of the current sample well, and we refer to the predictions as being consistent. (2) If the predictions are split evenly between two different outcomes, the predictions are classified as controversial. (3) In all other scenarios, we regard the prediction results of the model as confusing. States (2) and (3) are collectively referred to as inconsistent. Regarding the final prediction, we employ a voting strategy, that is, in states (1) and (3), we select the majority prediction as the final result. For case (2), the prediction made on the original input, namely (S_a, S_b), is used as the final result. Examples of the voting process can be seen in Figure 1. We call this method Infer with Rephrasing on Baseline model (Baseline-IWR) and the results are presented in the second row of Table 1. It is evident that the model exhibits poor consistency, with approximately 30% of the samples yielding inconsistent predictions. Furthermore, the accuracy rate on these samples substantially diminishes compared to that on the consistent samples (from 93.44% to 71.64%). This indicates that the model's semantic representation of these samples is inadequate, leading to difficulties in making accurate judgments.

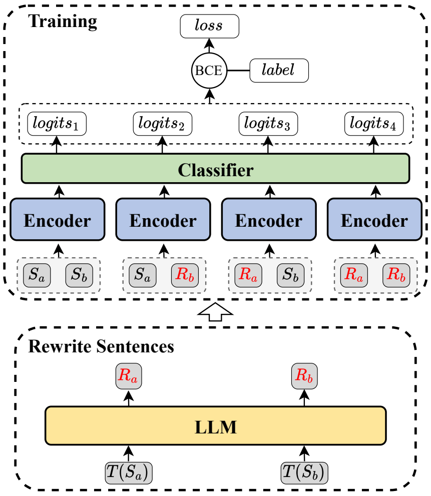

Figure 2: Framework of training.

Based on the aforementioned experimental results, we propose the training framework as shown in Figure 2. Specifically, for a sample X in the training set, we first expand it to the corresponding $\hat{X}$ using Formula (1). Then, the four pairs of texts in $\hat{X}$ are encoded simultaneously. It is worth noting that all these encoders share parameters. After

Model	Consistent		Inconsistent		Total Acc.
	P.(%)	Acc.(%)	P.(%)	Acc.(%)	
Baseline	-	-	-	-	86.70
Baseline-IWR	69.87	93.44	30.13	71.64	86.87
TIWR	89.58	90.97	10.42	64.08	88.17
TIWR-H	89.27	92.48	10.73	65.77	**89.62**
TIWR-P	89.58	90.97	10.42	73.98	89.20

Table 1: Comparison results on LCQMC test set. We categorize the predictions into two classes, namely Consistent and Inconsistent. P. and Acc. are used to denote the proportion of samples and the accuracy rate within the respective categories. Total Acc. represents the accuracy across the entire test dataset.

passing through the classification layer, the final loss is the sum of their respective losses:

$$logtis_i = \text{Classifier}(\text{Encoder}(\hat{X}_i)), \qquad (2)$$

$$loss = \sum_{i=1}^{4} \text{BCEloss}(logits_i, l), \qquad (3)$$

where $\hat{X}_i$ represents the i-th sample pair in $\hat{X}$, Classifier is a two-layer feed forward network and BCEloss refers to binary cross-entropy loss. This training method informs the model that different text representations share the same label, thereby encouraging the model to focus more on semantics rather than the words themselves. During inference, we adopt the same strategy as the Baseline-IWR. We refer to this method as Train and Infer with Rephrasing (TIWR). The experimental results are displayed in the third row of Table 1. The proportion of inconsistent predictions significantly decrease, moving from 30.13% to 10.42%, and the overall accuracy improves. The model performs worse on those samples with inconsistent predictions than Baseline-IWR (accuracy rate decreased from 71.64% to 64.08%). This can be primarily attributed to the fact that TIWR is more reliant on semantics for decision-making, thus enabling it to more accurately identify samples for which it cannot profoundly comprehend the semantics.

Evidently, there is considerable room for improvement on samples with inconsistent predictions. To address this, we employ the Baseline-IWR to make inferences on the training set, identifying the samples with inconsistent prediction, which we term hard cases. The remaining samples are classified as easy cases. Next, we combine a portion of hard cases with some easy cases as a new training set. Utilizing the TIWR as a warm start, we train a new model named TIWR-H. Subsequently, we propose two strategies: 1) Directly using TIWR-H for inference. 2) We construct a pipeline composed of TIWR and TIWR-H, named as TIWR-P. Specifically, we initially predict using the TIWR. If the results are consistent, we directly take this result as the final outcome. Otherwise, we further predict using TIWR-H. The results are shown in the last two rows of Table 1 respectively. By comparing TIWR and TIWR-P, it is evident that there is a significant improvement in accuracy on samples with inconsistent prediction (from 64.08% to73.68%). Moreover, both strategies achieve higher overall performance than TIWR.

4 Experiment

4.1 Datasets

Dataset	Train	Dev	Test
LCQMC	238,766	8,802	12,500
BQ	100,000	10,000	10,000
QQP	363,846	40,430	-
MRPC	3,668	408	1,725

Table 2: Statistics of datasets.

We evaluate our approach on two Chinese datasets, LCQMC (Liu et al., 2018) and BQ (Chen et al., 2018), and two English datasets, QQP (Shankar Iyer, 2012) and MRPC (Dolan and Brockett, 2005). The detailed information is listed in Table 2. LCQMC is a large-scale open-domain corpus for matching Chinese questions, while BQ is a domain-specific corpus for matching bank-related questions. Both of the two English datasets are corpora of sentence pairs automatically extracted from online websites.

4.2 Implementation Details

The LLM utilized in this paper is the chat version of Qwen (Bai et al., 2023), an open-source model with 14 billion parameters. We employ the base versions of BERT and RoBERTa respectively as encoders for the Chinese and English datasets. The weights for LLM and encoders are sourced from Hugging Face[1]. All results represent the average of five experimental trials. More detailed training information is in Appendix A.

4.3 Results and Analysis

Model	LCQMC	BQ
BERT† (Devlin et al., 2019)	86.75	85.17
GMN (Chen et al., 2020)	87.30	85.60
LET (Lyu et al., 2021)	88.38	85.30
DSSTM (Deng et al., 2022)	88.90	85.40
OTE (Ma et al., 2022)	88.29	85.26
CBM (Chen et al., 2022)	88.80	86.16
GBT(Peng et al., 2023)	89.20	-
TIWR†	87.83	86.01
TIWR-H†	**89.50**	85.75
TIWR-P†	88.93	**86.23**

Table 3: Experimental results on two Chinese datasets. We report the accuracy scores on their respective test sets. Methods with † indicate our implementation.

Model	QQP	MRPC
Roberta† (Liu et al., 2019)	91.6	87.2
-large version	92.0	87.6
DC-Match (Zou et al., 2022)	91.7	88.1
GBT(Peng et al., 2023)	**91.8**	-
TIWR†	91.6	88.3
TIWR-H†	91.6	88.4
TIWR-P†	91.6	**88.5**

Table 4: Experimental results on two English datasets. We report the accuracy scores on the QQP development set and MRPC test set.

The main results of comparison models on the Chinese and English datasets are presented separately in Table 3 and Table 4. We enumerate some of the state-of-the-art models from recent years. It can be seen that our model outperforms all these models on LCQMC, BQ, and MRPC. In QQP, our approach do not yield better results than the Baseline. We hypothesize that this is due to the lack of sufficient data within the training set to support the model in accomplishing certain semantic modeling. The comparative results among TIWR, TIWR-H, and TIWR-P further corroborate this point. This is because even if we select inconsistent samples from the training set for further training, it does not enhance the performance of the model.

4.4 Ablation Experiments

Model	LCQMC	BQ	QQP	MRPC
Baseline	86.8	85.2	91.6	87.2
DA	86.1	85.1	91.0	87.0
DA-IWR	86.1	85.3	91.1	87.0
TIWR-H/P	**89.5**	**86.2**	91.6	**88.5**

Table 5: Comparison of accuracy score between different models on various datasets.

While our original intention is not simply data augmentation, we indeed introduce additional data during the training of the TIWR series models. To measure the extent to which our experimental results are affected by data augmentation, we train the DA and DA-IWR models using a pure data augmentation paradigm. Besides incorporating additional data obtained via Formula 1 into the training set, their other configurations are respectively analogous to the Baseline and Baseline-IWR. The experimental results are shown in Table 5. It can be observed that the sole usage of data augmentation paradigms almost does not enhance the performance of the model, indicating that the model still fails to improve semantic modeling under such circumstances. Even more, the performance on certain datasets experience a slight decline. We postulate that this is due to the additional data not only failing to assist the models in better semantic modelling but also increasing the difficulty of learning superficial patterns.

5 Conclusion

This paper initially proposes a method to explore the degree to which PLMs model semantics and verifies that their ability to model semantics is relatively weak in STM tasks. Subsequently, we propose a training framework to enhance this ability of the model. We conduct experiments on two Chinese and two English datasets, validating the effectiveness of our method. Furthermore, it can be observed from the 'Inconsistent' column in Table 1 that there is still room for further improvement on these samples. Although the experiments in this paper are solely focused on STM tasks in NLU, theoretically, our method can easily be extended to various NLU tasks. This might be a direction that warrants further exploration.

[1] https://huggingface.co/

Limitations

Training: When generating data for training, Formula 1 cannot handle the noise caused by incorrect labels in the dataset. If a sample X has an incorrect label, then the labels of the four text pairs in $\hat{X}$ will also be incorrect. **Inference:** For the LCQMC dataset, the optimal model is TIWR-H. Therefore, during inference, we can entirely disregard the LLM for generating the corresponding $\hat{X}$ and only use X. We have verified through experiments that in this case, the accuracy can still reach 89.51%, which is almost identical to the standard TIWR-H. However, for the BQ and MRPC datasets, achieving the best results requires the use of the TIWR-P pipeline strategy, thus necessitating the use of LLM to generate the corresponding $\hat{X}$. This significantly increases the cost of inference. Additionally, for some samples, LLMs may refuse to rewrite sentences due to security policy reasons.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The BQ corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 4946–4951. Association for Computational Linguistics.

Lu Chen, Yanbin Zhao, Boer Lyu, Lesheng Jin, Zhi Chen, Su Zhu, and Kai Yu. 2020. Neural graph matching networks for chinese short text matching. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 6152–6158. Association for Computational Linguistics.

Mao Yan Chen, Haiyun Jiang, and Yujiu Yang. 2022. Context enhanced short text matching using clickthrough data. CoRR, abs/2203.01849.

Yao Deng, Xianfeng Li, Mengyan Zhang, Xin Lu, and Xia Sun. 2022. Enhanced distance-aware self-attention and multi-level match for sentence semantic matching. Neurocomputing, 501:174–187.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005. Asian Federation of Natural Language Processing.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC: A large-scale chinese question matching corpus. In Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, pages 1952–1962. Association for Computational Linguistics.

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. CoRR, abs/1907.11692.

Boer Lyu, Lu Chen, Su Zhu, and Kai Yu. 2021. LET: linguistic knowledge enhanced graph transformer for chinese short text matching. In Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, pages 13498–13506. AAAI Press.

Haoyang Ma, Zhaoyun Ding, Zeyu Li, and Hongyu Guo. 2022. OTE: an optimized chinese short text matching algorithm based on external knowledge. In Knowledge Science, Engineering and Management - 15th International Conference, KSEM 2022, Singapore, August 6-8, 2022, Proceedings, Part I, volume 13368 of Lecture Notes in Computer Science, pages 15–30. Springer.

Rui Peng, Zhiling Jin, and Yu Hong. 2023. GBT: generative boosting training approach for paraphrase identification. In Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, pages 6094–6103. Association for Computational Linguistics.

Kornél Csernai Shankar Iyer, Nikhil Dandekar. 2012. First quora dataset release: Question pairs.

Shufan Wang, Laure Thompson, and Mohit Iyyer. 2021. Phrase-bert: Improved phrase embeddings from BERT with an application to corpus exploration. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 10837–10851. Association for Computational Linguistics.

Sirui Wang, Di Liang, Jian Song, Yuntao Li, and Wei Wu. 2022. DABERT: dual attention enhanced BERT for semantic matching. In Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022, pages 1645–1654. International Committee on Computational Linguistics.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 4896–4907. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 1298–1308. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the AI ocean: A survey on hallucination in large language models. CoRR, abs/2309.01219.

Yicheng Zou, Hongwei Liu, Tao Gui, Junzhe Wang, Qi Zhang, Meng Tang, Haixiang Li, and Daniel Wang. 2022. Divide and conquer: Text semantic matching with disentangled keywords and intents. In Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 3622–3632. Association for Computational Linguistics.

A Implementation Details

The BERT and RoBERTa models used in this paper have 103 million and 125 million parameters, respectively. We apply AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$) with a weight decay rate of 0.01 for BERT and 0.1 for RoBERTa and use a warm-up of 0.1 for learning rate decay. For the baseline and TIWR model, we use an epoch of 3 and a batch size of 16. The learning rate is set to be 2e-5 for MRPC and 3e-5 for other datasets. As for the TIWR-H model, the epoch is set to be 3 for QQP and 1 for other datasets. The learning rate is selected in {2e-6, 3e-5}. We attempt various mix ratios of hard cases and easy cases to construct the training set required for TIWR-H. Specifically, we define α as the proportion of hard cases in the training set, and experiment with several distinct values, namely $\{0.2, 0.4, 0.6, 0.8, 1\}$. Ultimately, for LQCMQ, QQP, and MRPC, the values of α are set at 0.8, 0.2, and 0.6 respectively. For BQ, none of these mixtures yield satisfactory results, thus we adopt an alternate approach in which only the positive samples from the easy cases are used for mixing. All experiments are performed utilizing an A100 40G GPU.

B Proportion of Inconsistent Predictions

Figure 3 presents a comparison of the proportion of inconsistent predictions between Baseline-IWR and TIWR on different datasets. It can be seen that the number of samples with inconsistent predictions decreases significantly across all datasets.

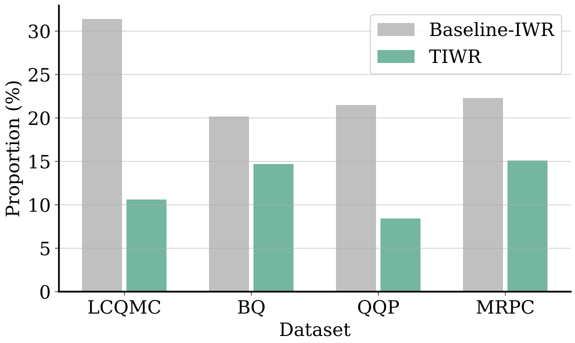

Figure 3: Changes in the proportion of inconsistent predictions.