YNU-HPCC at SemEval-2024 Task 1: Self-Instruction Learning with Black-box Optimization for Semantic Textual Relatedness

Weijie Li, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
liweijie01@stu.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper introduces a system designed for SemEval-2024 Task 1 that focuses on assessing Semantic Textual Relatedness (STR) between sentence pairs, including its multilingual version. STR, which evaluates the coherence of sentences, is distinct from Semantic Textual Similarity (STS). However, Large Language Models (LLMs) such as ERNIE-Bot-turbo, typically trained on STS data, often struggle to differentiate between the two concepts. To address this, we developed a self-instruction method that enhances their performance distinguishing STR, particularly in cases with high STS but low STR. Beginning with a task description, the system generates new task instructions refined through human feedback. It then iteratively enhances these instructions by comparing them to the original and evaluating the differences. Utilizing the Large Language Models' (LLMs) natural language comprehension abilities, the system aims to produce progressively optimized instructions based on the resulting scores. Through our optimized instructions, ERNIE-Bot-turbo exceeds the performance of conventional models in Track A,achieving a score enhancement of 4 to 7% on multilingual development datasets.

1 Introduction

SemEval-2024 Task 1(Ousidhoum et al., 2024) addresses the challenge of Semantic Textual Relatedness (STR), which goes beyond paraphrasing and entailment of Semantic Textual Similarity (STS)(Agirre et al., 2012, 2016; Cer et al., 2017; Xu et al., 2015) by considering topics and logical connections between sentence pairs. This task is particularly complex due to the nuanced context required for STR, a feature not fully captured by existing models trained predominantly on STS data. This gap can lead to black-box Large Language Models (LLMs) misinterpretations like ERNIE-Bot-turbo[1].

Our study introduces a self-instruction method to enhance the distinction between STS and STR in LLMs(Chen et al., 2023; Zhang et al., 2023; Hou et al., 2022; Wei et al., 2021). In our approach, back translation(Sennrich et al., 2016) converts low-resource language sentence pairs into English as inputs for LLMs. With a task description as the starting point, the black-box LLMs generate a new task instruction, which will be refined based on human feedback. The system iteratively refines the enhanced instruction by assessing it against the original and using the resulting score to produce increasingly optimized instructions. Our method improves how LLMs deal with tricky cases of similar but unrelated texts. Using our optimized instructions, ERNIE-Bot-turbo outperforms standard models and boosts scores by 4 to 7% on multilingual development datasets in Track A. The ranking of each Tack A's test dataset is as follows: English (36), Amharic (11), Algerian Arabic (24), Telugu (24), Spanish (24), Moroccan Arabic (24), Marathi (25), Kinyarwanda (20), and Hausa (20). The remainder of this paper is organized as follows. Section 2 describes the model and method used in our system, Section 3 discusses the results of the experiments, and finally, conclusions are drawn in Section 4.

2 Methodology

Figure 1 illustrates the overall framework of our self-instruction method. We employ back translation for datasets encompassing multiple languages to render sentence pairs into English as the input for LLMs. With a task description as the starting point, the black-box LLMs generates a new task instruction which will be refined based on human feedback. The enhanced instruction is subsequently assessed against the original, generat-

[1]https://yiyan.baidu.com/

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 792–799

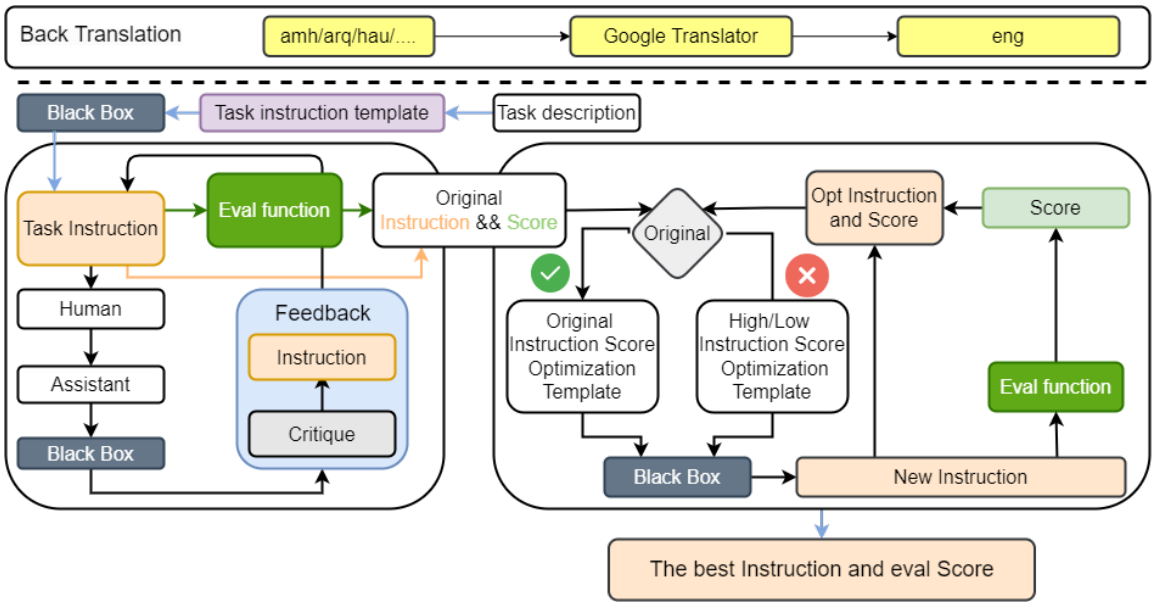

Figure 1: The framework of self-instruction method

ing a score that informs the optimization cycle. The system progressively refines the instructions in response to this score, resulting in progressively more optimized directives. The ensuing section will delve into a detailed analysis of this iterative optimization process.

2.1 Design of task instruction

Using sentences from the Amharic dataset as examples of hard sample with high Semantic Textual Similarity (STS) yet subtle Textual Relatedness (STR) for instance, What made him so certain? What contributed to his happiness? (original Amharic: "**ይህን ያህል እርግጠኛ እንዲሆን ያደረገው ምንድን ነው? ደስተኛ እንዲሆን አስተዋጽኦ ያደረገው ምንድን ነው?**"; goal label: 0.39) we underscore the significance of three components: instruction, the Chain of Thought (CoT)(Wei et al., 2022), and easily confused examples(Zhang et al., 2022; Li and Qiu, 2023). Human generated instructions aid LLMs in grasping the primary task but may not adequately explicate the concept of semantic textual relatedness (Figure 2.a) (Pred Score: 0.83). The CoT process facilitates LLMs in logical reasoning and analyzing sentence pairs, yet it encounters obstacles with complex samples prone to creating illusions (Figure 2.b) (Pred Score: 0.77). Easily confused examples are practical in dissecting hard samples but can skew the assessment of standard samples (Figure 2.c) (Pred Score: 0.67). Consequently, merging these approaches could provide more practical guidance for LLMs in discerning the relatedness of sentence pairs (Figure 2.d) (Prediction Score: 0.35). Detailed findings from the ablation study are discussed in Section 3.

2.2 Two fundamental components to generate the task instruction

Making use of natural language task description. LLMs excel in understanding natural language and simplifying the definition of optimization tasks. Capitalizing on this, we employ LLMs to convert the initial task description into detailed task instruction, guiding the LLMs to perform tasks such as STR analysis effectively, as indicated in Figure 3.a.

Refining task instruction through human feedback and evaluating their performance. While Large Language Models (LLMs) can generate task instructions from description, these instructions often fall short of being optimal and thus require human refinement and critical feedback. For instance, LLMs may overlook the significance of high and low relatedness (Figure 3.b). Subsequently, the improved instructions are evaluated, and their scores and instructions are integrated into the original framework, streamlining the subsequent optimization process (Figure 3.c).

2.3 LLMs as the black-box optimizer

After obtaining the original instruction-score (Figure 4.a), we utilize LLMs as the black-box optimizer to update and optimize the instructions iteratively. In each optimization step, the optimizer

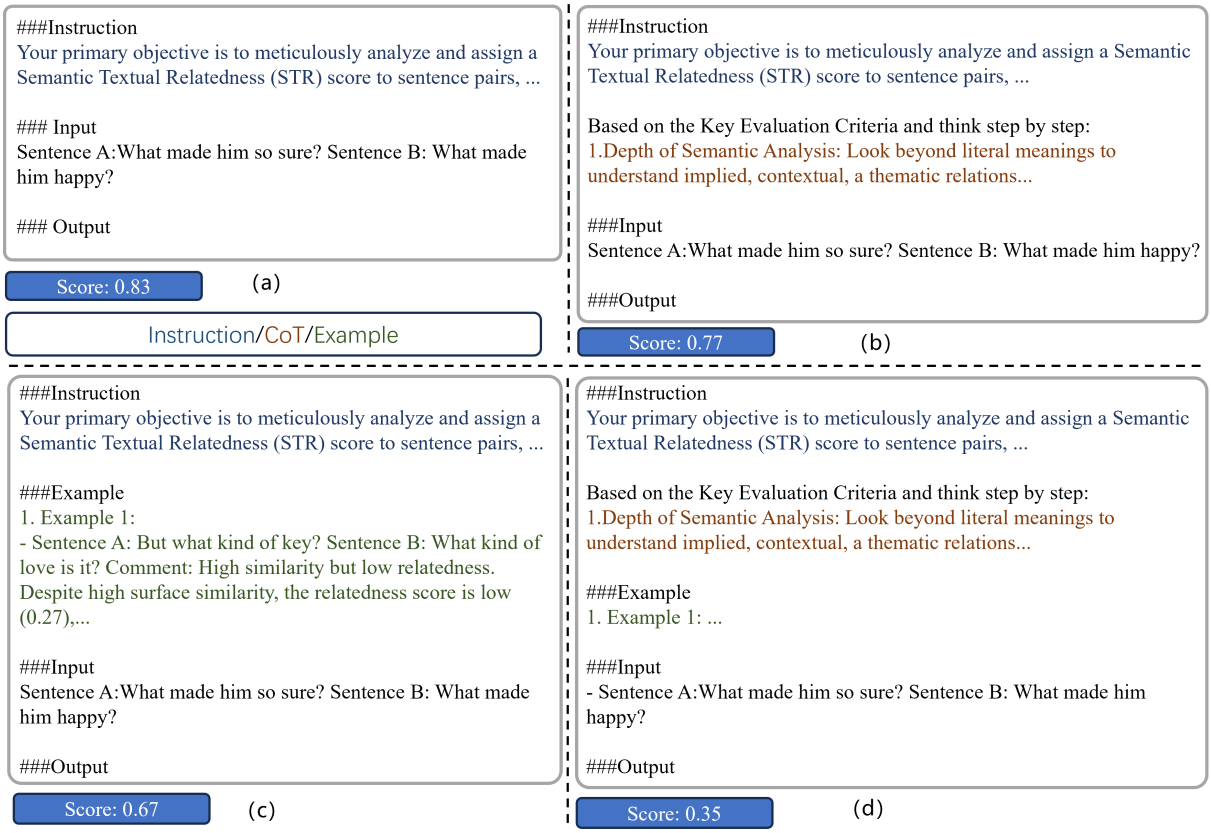

Figure 2: The process of task instruction design. (a)instruction(b)instruction + chain of thought(c)instruction + easily confused examples(d)instruction + chain of thought + easily confused examples

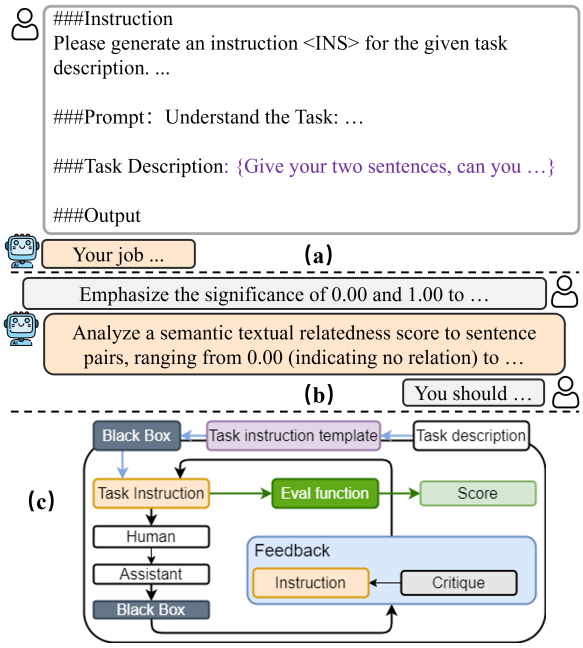

Figure 3: The process of original instruction optimization:(a)task instruction(b)task instruction optimization (c)overall process

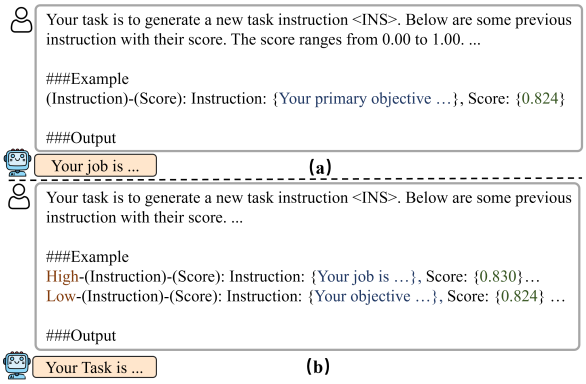

Figure 4: The process of utilizing LLM as the black-box optimizer

LLM generates candidate optimal instructions by analyzing the strengths of high-scoring and the weaknesses of low-scoring instructions (Figure 4.b). Subsequently, the new instruction is evaluated and added to the instruction-score list for sorting. From the instruction-score list, the top five high-scoring and the bottom five low-scoring instructions are selected and added to the instruction optimization template. The optimization process continues until the LLMs cannot propose new solutions with better optimization scores or the maximum number of optimization steps is reached.

Table 1: The evaluation scores of representative models from the four model methods on the training set.

BERT		Dual Sentence Encoding	
Model	Score	Model	Score
bert-base-uncased	0.673	all-mpnet-base-v2	0.787
bert-large-uncased	0.609	**all-MiniLM-L6-v2**	**0.824**
distilbert-base-uncased	0.673	all-MiniLM-L12-v2	0.816
deberta-base	0.668	all-distilroberta-v1	0.802
deberta-large	**0.678**	sentence-t5-base	0.805
deberta-large-mnli	0.659	sentence-t5-large	0.81
deberta-xlarge-mnli	0.651	sentence-t5-xl	0.805
distilroberta-base	0.618	moco-sentencebertV2.0	0.797
roberta-base	0.635		
roberta-large	0.44		
roberta-large-mnli	0.439		
Contrastive Learning		**LLM**	
Model	Score	Model	Score
sup-SimCSE-VietNamese-phobert-base	0.64	t5-base	0.705
sup-simcse-roberta-large	0.743	t5-large	0.702
sup-simcse-roberta-base	0.744	flan-t5-base	0.665
sup-simcse-bert-base-uncased	**0.8**	flan-t5-large	0.679
unsup-simcse-roberta-large	0.769	ERNIE-Bot-turbo(w/o opt)	0.782
diffcse-bert-base-uncased-sts	0.783	**ERNIE-Bot-turbo(w/ opt)**	**0.883**
diffcse-bert-base-uncased-trans	0.761		
diffcse-roberta-base-sts	0.774		
diffcse-roberta-base-trans	0.78		
esimcse-bert-base-uncased	0.778		
esimcse-bert-large-uncased	**0.798**		
esimcse-roberta-base	**0.792**		
esimcse-roberta-large	0.764		
pcl-bert-base-uncased	0.776		
pcl-bert-large-uncased	**0.799**		
pcl-roberta-base	0.766		
pcl-roberta-large	0.755		

3 Experimental Result

Datasets. The STR task dataset comprises datasets in 14 distinct languages, including 9 languages specifically for Track A. Each language dataset contains pairs of sentences, where each pair in the training, development, and test sets is assigned a gold score. This score reflects the degree of STR between the two sentences, ranging from 0 to 1, as determined by manual annotation. Figure 5 below presents the composition of the training, test, and development sets for Track A.

Evaluation Metrics. The STR in Track A is evaluated using the spearman rank correlation coefficient(Sedgwick, 2014), which measures how well the system predicted rankings of test instances align with human judgment. The metric will be calculated as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \quad (1)$$

where d_i represents the difference between the ranks of the i-th pair of sentences,n is the number of pairs of sentences,ρ is the spearman rank correlation.

3.1 Implementation Details

Our approach, addressing the scarcity of low-resource languages, uses back translation to convert their sentence pairs into English for (LLMs) inputs. This experiment prioritizes scoring on English datasets to select the most effective score model. We assess four baseline methods: BERT(Devlin et al., 2019; Sanh et al., 2019; He et al., 2020; Delobelle et al., 2020; Raffel et al., 2020; Chung et al., 2022), dual sentence encoding(Reimers and Gurevych, 2019; Ni et al., 2022), contrastive learning(Gao et al., 2021; Song et al., 2020; Wang et al., 2020; Chuang et al., 2022; Wu et al., 2022b,a) , and LLMs. These models were evaluated using the training set, with results presented in Table 1. Considering our experimental objective of analyzing hard samples and scoring sentence pair STR, ERNIE-Bot-turbo was chosen as the scoring model. The LLMs utilized as the optimizer and scorer are: (a)optimizer LLM: gpt-3.5-turbo and (b)scorer LLM: ERNIE-Bot-turbo.

3.2 Desigin of task instruction

At the experiment's outset, we performed adaptation tests on the English training dataset using four variations of instruction templates: (1) instruction only, (2) instruction with chain-of-thought, (3) instruction with easily confused examples, and (4) instruction with both chain-of-thought and easily confused examples. The experimental results in Figure 6 suggest that combining instruction, chain-of-thought, and easily confused examples significantly aids LLMs in semantic textual relatedness analysis.

3.3 Prompt optimization

The score LLM operates at a temperature of 0, ensuring deterministic decoding, whereas the optimizing LLM uses a temperature of 0.95 promoting creativity in instruction generation. Figure 7.a illustrates the accuracy fluctuations during the model's evaluation on the English training dataset. Figure 7.b presents the scores for Track A's development in three scenarios: without optimization, optimized (val-score: 0.8360) and further optimized (val-score: 0.8839). Figure 8 delves into the impact of these three optimization scenarios on hard samples. Consequently, our methodology effectively reduces the hallucinations of LLMs in STS and STR tasks. This leads to a more comprehensive analysis of hard samples and consistently

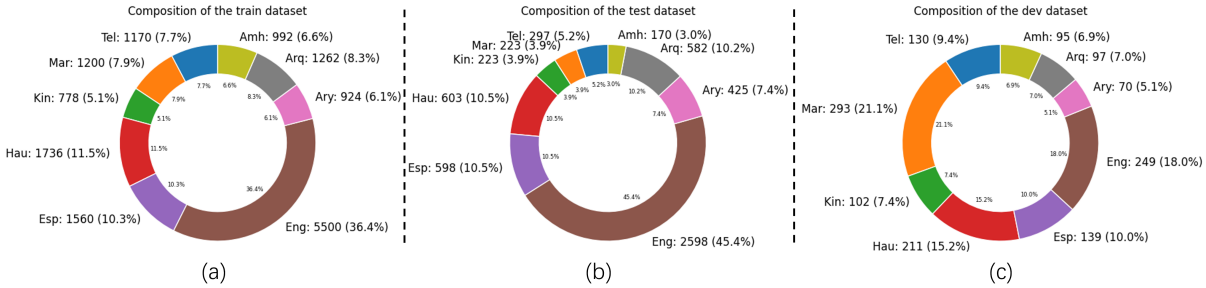

Figure 5: The composition of the Track A's training(a), test(b), and development(c) dataset

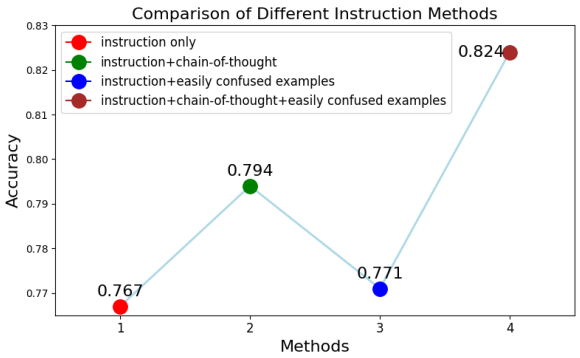

Figure 6: The ablation tests for four variations of instruction templates: 1.instruction 2. instruction + chain of thought 3.instruction + easily confused examples 4.instruction + chain of thought + easily confused examples

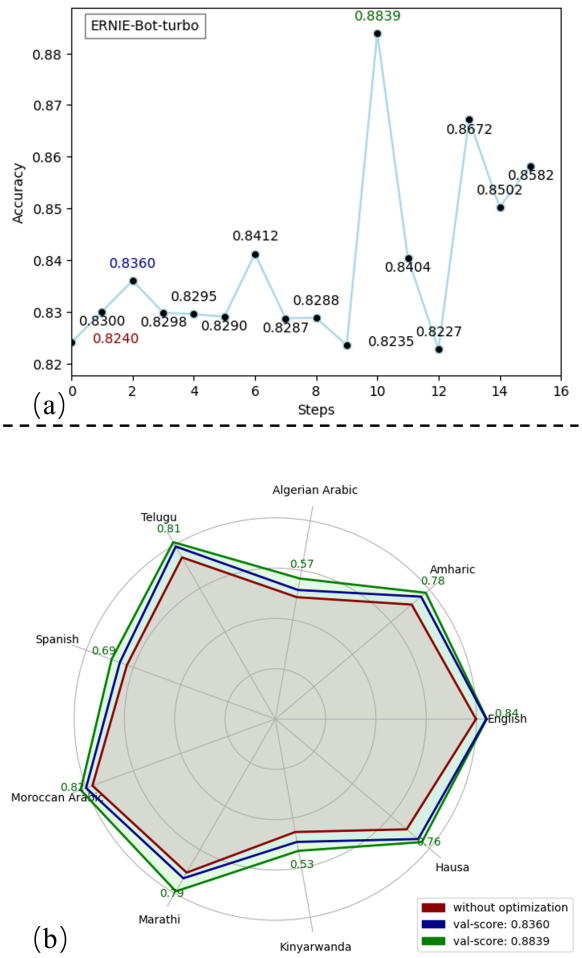

Figure 7: (a) shows changes in accuracy during evaluation on the English training dataset. (b) shows the development scores for Track A in scenarios: without optimization, optimized (val-score: 0.8360), and further optimized (val-score: 0.8839).

improves performance evaluations on the training dataset through an iterative process.

3.4 Result and Discussion

Results. Our final evaluation compared the 'no optimization' approach to 'optimization' across Track A's nine language development datasets using back translation, as shown in Figure 9.The outcomes indicate that optimized instructions significantly enhanced performance by 4 to 7% over the non-optimized approach.The ranking of each test dataset are as follows: English (36), Amharic (11), Algerian Arabic (24), Telugu (24), Spanish (24), Moroccan Arabic (24), Marathi (25), Kinyarwanda (20), and Hausa (20).

Discussion. The experimental results suggest the following:

- Our self-instruction method effectively reduces confusion between STS and STR in Large Language Models (LLMs), thereby improving accuracy and enhancing the LLMs's capability to analyze standard samples, particularly in examining hard sample.
- However, the experimental outcomes are somewhat modest due to the coarse granularity of the STR task and the considerable overlap between semantic textual similarity and relatedness.

##Instruction
Your task is to predict a relatedness score between two sentences based on their semantic relatedness. …

Based on the Key Evaluation Criteria and think step by step:
1. Depth of Semantic Analysis: ...

###Examples and Analysis:
1. Example 1:SentenceA:But what kind of key? SentenceB:What kind of love is it?
Comment:Highly similar but low relatedness (Relatedness score: 0.27).
2. Example 2: ...

###Input
SentenceA: What made him so sure? Sentence B： What made him happy?

###Output

Score: 0.67

(without any optimization)

##Instruction
Our responsibility is to meticulously analyze and assign a Semantic Textual Relatedness (STR) score between sentence pairs, …

Based on the Key Evaluation Criteria and think step by step
1.Depth of Semantic Analysis: ...
2.Significance of Relatedness: ...

###Examples and Analysis
1. Example 1:
-SentenceA:But what kind of key? SentenceB:What kind of love is it?
Comment:(Highly similarity,but low relatedness.the relatedness score is 0.27)
2. Example 2: ...

###Input
SentenceA: What made him so sure? SentenceB: What made him happy?

###Output

Score: 0.55 **(val-score : 0.8360)**

##Instruction
Your primary objective is to meticulously analyze and assign a Semantic Textual Relatedness (STR) score to sentence pairs, ranging from 0.00 ...
Emphasize the significance of differentiating ...

Based on the Key Evaluation Criteria and think step by step:
1. Depth of Semantic Analysis: ...
2. Significance of Relatedness: ...
1.Low Relatedness: ...

###Examples and Analysis:
1. Example 1:
SentenceA:But what kind of key? SentenceB:What kind of love is it?Comment: Despite high surface similarity, the relatedness score is low (0.27), indicating a lack of substantial semantic connection...
...

###Input
SentenceA: What made him so sure? Sentence B： What made him happy?

###Output

(val-score: 0.8839)

Score: 0.42

Figure 8: It demonstrates how the scoring model assesses the impact of these three optimization scenarios on hard samples.(red: score, brown: chain-of-thought optimization,blue: example analysis optimization, purple: instruction optimization)

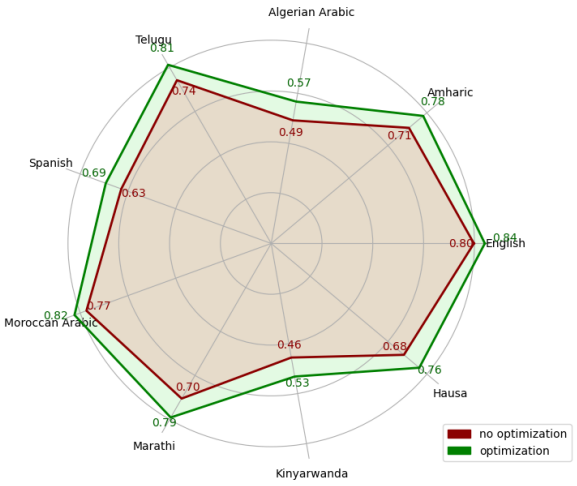

Figure 9: The performance of no optimization and optimization in development dataset.

- The back translation method encounters notable challenges when utilized with low-resource languages such as Arabic. This is primarily due to significant language biases between low-resource and high-resource languages like English within the semantic space, directly influencing the scoring model's judgment.

- The limitation of the score model is still an obstacle to performance. ERNIE-Bot-turbo (score model), trained on Chinese and English datasets corpus, demonstrates weaker proficiency in evaluating English sentence pairs.

4 Conclusion

In this paper, we developed a self-instruction method that enhances LLMs' ability to distinguish between Semantic Textual Similarity (STS) and Semantic Textual Relatedness (STR), particularly in hard samples(High STS but low STR).Through this method, ERNIE-Bot-turbo (score LLM) not only surpasses the performance of conventional models, achieving a score enhancement of 4 to 7 % on multilingual development datasets, but also effectively reduces confusion between STS and STR in Large Language Models (LLMs). Additionally, it achieved a commendable ranking in the final test evaluation. Our work demonstrates that optimized instructions, chain of thought, and easily confused examples enable LLMs to mitigate errors even in few-shot samples. Future research will aim to refine LLMs' capacity to grasp the overall semantic meaning of sentences further.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments.This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos.61966038 and 62266051.

References

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2023. Instructzero: Efficient instruction optimization for black-box large language models. *arXiv preprint arXiv:2306.03082*.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4207–4218, Seattle, United States. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. 2022. MetaPrompting: Learning to learn better prompts. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3251–3262, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Nedjma Ousidhoum, Shamsuddeen Hassan Muhammad, Mohamed Abdalla, Idris Abdulmumin, Ibrahim Said Ahmad, Sanchit Ahuja, Alham Fikri Aji, Vladimir Araujo, Abinew Ali Ayele, Pavan Baswani, et al. 2024. Semrel2024: A collection of semantic textual relatedness datasets for 14 languages. *arXiv preprint arXiv:2402.08638*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Philip Sedgwick. 2014. Spearmans rank correlation coefficient. *Bmj*, 349.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022a. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. *arXiv preprint arXiv:2201.12093*.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhihan Zhang, Shuohang Wang, Wenhao Yu, Yichong Xu, Dan Iter, Qingkai Zeng, Yang Liu, Chenguang Zhu, and Meng Jiang. 2023. Auto-instruct: Automatic instruction generation and ranking for black-box language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9850–9867, Singapore. Association for Computational Linguistics.