An Empirical Study of Prompt-based Non-functional Requirements Classification

Xia Li

The Department of Software Engineering and Game Design and Development,
Kennesaw State University,
Marietta, USA

Abstract. In modern software development, Non-Functional Requirements (NFR) are essential to satisfy users' needs, which define various constraints and qualities that the system must adhere (e.g., quality, usability, security). Since NFR play a critical role in the guidance of architectural design, it is important to extract different NFR from software requirements specification documents early and accurately. However, distinguishing different categories of NFR is tedious, error-prone, and time-consuming due to the complexity of software systems. In our paper, we conducted a comprehensive study to evaluate the performance of prompt-based non-functional requirements classification by designing various handcraft templates and soft templates on the pre-trained language model (i.e., BERT). Our experimental results show that handcraft templates can achieve best effectiveness (e.g., 83.52% in terms of F1 score) but with unstable performance for different templates. Also, the performance can become stable after soft templates are concatenated with handcraft templates. For example, the standard deviation of F1 score for four combined templates can be improved to 0.74 from 1.00 for handcraft templates.

Keywords: Non-functional requirements classification, prompt-based learning, pre-trained models

1 Introduction

In modern software development, Non-Functional Requirements (NFR) are essential to satisfy users' needs, which define various constraints and qualities that the system must adhere (e.g., quality, usability, security). Since NFR play a critical role in the guidance of architectural design, it is important to extract different NFR from software requirements specification documents early and accurately. However, developers always overlook the importance of NFR since they tend to be across various requirement specification documents, making it difficult to locate and consolidate them effectively. Also, distinguishing different categories of NFR is tedious, error-prone, and time-consuming due to the complexity of software systems. Thus, the task of NFR classification is crucial for the whole software development process.

Due to the practical benefits of NFR classification, researchers have utilized various techniques to classify NFR and achieved impressive performance. For example,

David C. Wyld et al. (Eds): NLAI, IBCOM, MLDS, GridCom, SCAI, ICCSEA, SPPR – 2024
pp. 229-238, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142217

EzzatiKarami et al. [6] used various machine learning algorithms (e.g., Support Vector Machine, Decision Tree) by combining three feature extraction techniques (POS tagging, BoW, and TF-IDF) for NFR classification. Navarro-Almanza et al. [17] proposed to use deep learning models (e.g., Convolutional Neural Network (CNN)) to improve the performance of NFR classification. Recently, pre-trained foundation models (e.g., BERT [8], GPT [11]) have been widely used in various AI fields such as natural language processing (NLP), computer vision (CV) and graph learning (GL), which can be applied to many downstream tasks such as text classification [9] and image classification [10]. More promising works about pre-trained model can be found in the survey paper [7]. Pre-trained models are also applied into the field of requirements classification. For example, recent work [14] proposed to combine original requirement text with standard prompt templates (e.g., *This is requirement*) as the input sequence of the pre-trained model. Current survey on prompt engineering [15] demonstrates that different prompts can affect the performance of the pre-trained model so that it is necessary to evaluate the impact of different prompt templates on NFR classification. Furthermore, the study [12] also indicates that a learnable tensor can be concatenated with the input embeddings to become a series of soft templates for natural language understanding. Such soft templates can be also applied into prompt-based NFR classification. In this paper, we conduct a comprehensive study by designing various prompt templates (including handcraft templates and soft templates) for NFR classification based on pre-trained BERT model. Our study indicates that handcraft templates can achieve best effectiveness (e.g., 83.52% in terms of F1 score) but with unstable performance for different templates. Also, the performance can become stable after learnable templates (a.k.a., soft templates) are inserted with handcraft templates. This paper makes the following contributions:

- **Study**. A comprehensive study on NFR classification based on various templates.
- **Guidance**. The results can provide guidance for other prompt-based NFR classification techniques.

The structure of the paper is as follows. In Section 2, we introduce the related studies about software requirements classification. In Section 3, we illustrate how we conduct our study in the paper. In Section 4 and Section 5, we demonstrate our experimental settings and results analysis, respectively. We discuss the threats to validity in Section 6 and conclude our paper in Section 7.

2 Related Work

In this section, we discuss some related studies of requirements classification (including both functional requirements and non-functional requirements) via machine learning, deep learning and pre-trained models.

2.1 Requirements classification through traditional machine learning techniques

Traditional machine learning techniques have been widely used in requirements classification. Abad et al. [16] apply several machine learning methods (e.g., Biterm Topic Modeling, or Naive Bayes) through preprocessing and unifying dataset for both functional and non-functional requirements classification. Amasaki et al. [5] use vectorization methods (e.g., document embedding methods) and four supervised classification (e.g., Logistic Regression, Naive Bayes, Random Forests and SVM) for NFR Classification. Binkhonain et al. [4] compare the effectiveness of various machine learning models for NFR classification, showing that Support Vector Machines (SVMs) can help achieve the best performance on small NFR dataset.

2.2 Requirements classification through deep learning techniques

Due to the limitation of traditional machine learning techniques, deep learning is applied into requirements classification by many researchers. Navarro-Almanza et al. [17] utilize CNN to classify the 12 NFR categories. Dekhtyar et al. [18] use Word2Vec embeddings and CNN for NFR classification. Rahimi et al. [19] use ensemble approaches with a combination of four different deep learning models (e.g., LSTM, BiLSTM, GRU, and CNN) for more accurate requirements classification.

2.3 Requirements classification through pre-trained models

Powerful pre-trained models are applied into requirements classification based on the theory of transfer learning. Hey et al. [19] propose NoRBERT to fine-tune BERT model and apply it to different tasks for requirements classification, achieving promising performance. Chatterjee et al. [21] utilize a tool to automatically label a dataset from various software requirement specification documents and classify the new data via BERT model. Luo et al. [14] propose PRCBERT, an approach of prompt learning for requirement classification using BERT model that applies flexible prompt templates to achieve accurate requirements classification.

In this paper, we focus on non-functional requirements classification since they represent similar criteria that define how a system should behave, such as performance, security, usability, reliability, and scalability, which are not easily distinguished.

3 Study Approach

In this section, we introduce how we design our study. In detail, we introduce the general approach of the study in Section 3.1 and template designs in Section 3.2.

3.1 Overall of the study

In many classification tasks, the pre-trained models are typically used to generate a final vector representation for input sequence and an additional neural network is connected with the pre-trained model, leading to a low correlation between the input sequence and the target task. In this section, we introduce the overall structure of our study as shown in Figure 1 (inspired by the current work [14]). Based on the original requirement text, we design various templates (including a masked target label) that can be as the input of pre-trained models. We use pre-trained BERT model in this paper. During the training process, pre-trained models can predict the masked target label and the training loss is calculated for back propagation to fine-tune the pre-trained model by updating the parameters. We use cross-entropy loss function in our study since NFR classification is the classic multi-class classification problem.

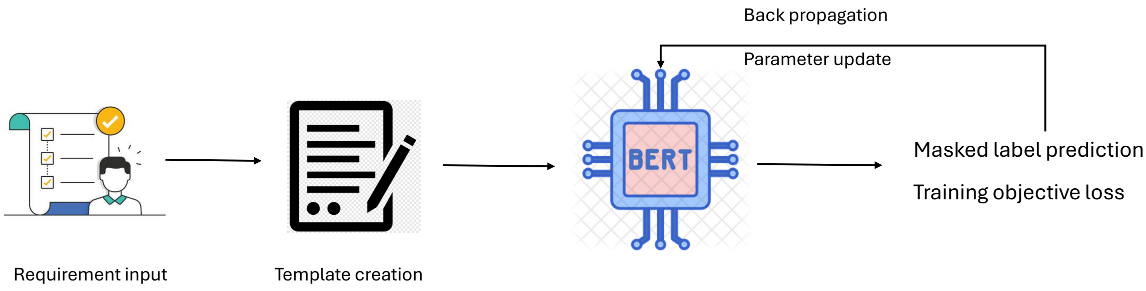
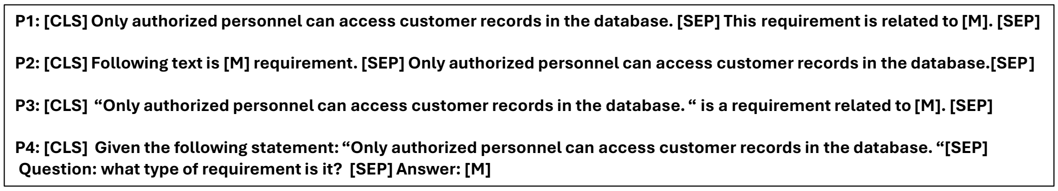


Fig. 1. Overall structure of the study

3.2 Template design

The major part of this paper is the evaluation of various templates on prompt-based NFR classification. In this section, we introduce how we design different prompt templates according to the two basic studies (i.e., PET [13] and p-tuning [12]). We use the security requirement “Only authorized personnel can access customer records in the database” as the example.

The study Pattern-Exploiting Training (PET) [13] is proposed to illustrate that a handcraft prompt can be appended after the input sequence and the target task is masked so that pre-trained model can predict the masked label then. This technique is able to bridge the gap between pre-trained models and specific downstream classification task. Recent study [14] also indicates that such handcraft prompts can help achieve promising performance in software requirements classification. However, the



P1: [CLS] Only authorized personnel can access customer records in the database. [SEP] This requirement is related to [M]. [SEP]

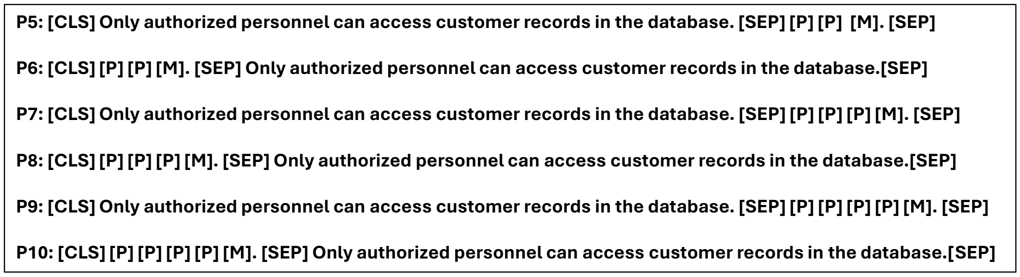
P2: [CLS] Following text is [M] requirement. [SEP] Only authorized personnel can access customer records in the database.[SEP]

P3: [CLS] “Only authorized personnel can access customer records in the database. “ is a requirement related to [M]. [SEP]

P4: [CLS] Given the following statement: “Only authorized personnel can access customer records in the database. “[SEP]
Question: what type of requirement is it? [SEP] Answer: [M]

Fig. 2. Handcraft templates

effectiveness of different handcraft prompts on NFR classification is not evaluated since prompt engineering shows that pre-trained models (e.g., GPT,BERT) can be affected significantly by different prompts. Thus, we apply PET by designing various handcraft templates and inserting the templates in different positions of the input requirement text. The designed templates are shown in Figure 2. Please note that [CLS] in the template represents a special token in BERT model [8] in the front of the original input text and [SEP] is a separator token to represent the segment of each sentence. [M] is the masked token to represent the requirement category (e.g., performance, security, usability) that can be predicted by BERT model.



P5: [CLS] Only authorized personnel can access customer records in the database. [SEP] [P] [P] [M]. [SEP]

P6: [CLS] [P] [P] [M]. [SEP] Only authorized personnel can access customer records in the database.[SEP]

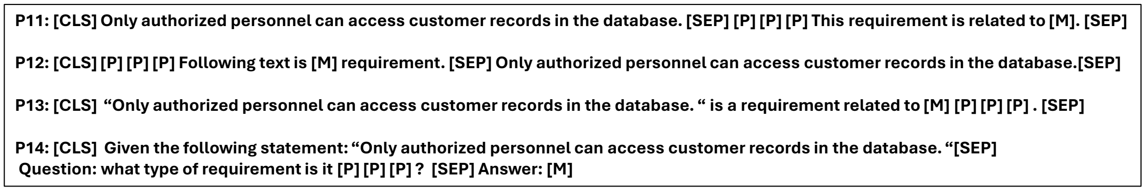
P7: [CLS] Only authorized personnel can access customer records in the database. [SEP] [P] [P] [P] [M]. [SEP]

P8: [CLS] [P] [P] [P] [M]. [SEP] Only authorized personnel can access customer records in the database.[SEP]

P9: [CLS] Only authorized personnel can access customer records in the database. [SEP] [P] [P] [P] [P] [M]. [SEP]

P10: [CLS] [P] [P] [P] [P] [M]. [SEP] Only authorized personnel can access customer records in the database.[SEP]

Fig. 3. Soft templates



P11: [CLS] Only authorized personnel can access customer records in the database. [SEP] [P] [P] [P] This requirement is related to [M]. [SEP]

P12: [CLS] [P] [P] [P] Following text is [M] requirement. [SEP] Only authorized personnel can access customer records in the database.[SEP]

P13: [CLS] “Only authorized personnel can access customer records in the database. “ is a requirement related to [M] [P] [P] [P] . [SEP]

P14: [CLS] Given the following statement: “Only authorized personnel can access customer records in the database. “[SEP]
Question: what type of requirement is it [P] [P] [P] ? [SEP] Answer: [M]

Fig. 4. Combination of handcraft and soft templates

Based on another type of prompt-based classification technique p-tuning [12], manual or handcraft prompt is discrete and often leads to unstable performance. In this case, learnable continuous prompt embeddings (soft templates) in concatenation with input requirement sequence can be created for better NFR classification. We evaluate such soft templates in the following two approaches. First, we insert different number of continuous tokens directly before or after the input requirement text without any handcraft prompts, shown in Figure 3. In this type of template design, [P] represents the learnable token that replaces the concrete templates in Figure 2. We evaluate different numbers of learnable tokens (e.g., 2, 3 and 4) and different positions in the input sequence (in front or back of the requirement text). However, even the continuous prompts can be learned by pre-trained model as described in p-tuning, the disadvantage is that such token is meaningless without context so that it is not easy to learn and accurately predict the masked label. To resolve this problem and stabilize the training performance, we combine both handcraft templates and soft template to build a comprehensive prompt for the pre-trained model, shown in Figure 4. In the template, we insert learnable tokens [P] in concatenation with the handcraft templates to create new ones.

4 Experimental Design

In this section, we introduce the dataset used in our study in Section 4.1 and the experimental configuration in Section 4.2.

4.1 Dataset

In our study, we use the widely used pre-labeled dataset PROMISE [3] with 914 non-functional requirements consisting of the following five categories: maintainability, operability, performance, security, and usability. Before creating different prompt templates, we pre-process the dataset using popular natural language processing steps such as stemming, lemmatization, stop-word removal and conversion to lower case via the widely used NLTK [2] toolkit. We also remove special characters that are unique in different domains.

4.2 Experimental configuration

We use the pre-trained foundation model BERT-base that can be downloaded from the popular AI hub Hugging Face [1]. For the hyperparameters, we set the maximum input sequence length as 256, batch size as 8, learning rate as $5e^{-5}$, epochs as 32. We also use AdamW optimizer [22] in the training process. We use the popular evaluation metrics precision, recall and F1 score for classification problems. We split the original dataset into training set (80%) and test set (20%). We also apply 10-fold cross-validation for each template introduced in Section 3.2. All training and inference steps are executed on a machine with Intel Core 13900K CPU, 32GB memory and NVIDIA RTX 4090 GPU.

5 Results Analysis

In this paper, we will investigate the following two research questions:

- **RQ1:** How does standalone handcraft templates and learnable soft templates affect the performance of NFR classification?
- **RQ2:** How does the combination of handcraft and learnable templates affect the performance of NFR classification?

5.1 Performance of handcraft templates and learnable soft templates

In this RQ, we investigate the performance of handcraft templates and learnable soft templates separately. Table 1 shows the results of NFR classification based on the 4 handcraft templates and 6 learnable soft templates in terms of the evaluation metrics precision, recall and F1 score. Please note that all results are calculated as the average values of 10-fold cross-validation based on each template. From the results, we have the following two findings. First, the overall performance of learnable soft templates are worse than handcraft templates for all metrics. For example, in terms of F1 score, the best result of learnable templates is 78.79% while the best result of handcraft templates is 83.52%. The possible reason is that there are no meaningful context for the special tokens [P] in the learnable soft templates so that it is not easy to predict the target label accurately. Second, even the handcraft template can achieve better results, the standard deviation of the four templates (1.00) is larger than learnable templates (0.84), showing unstable results for random handcraft templates. It motivates us to combine both templates for more stable and accurate NFR classification.

5.2 Effectiveness of handcraft templates and learnable soft templates

In this RQ, we combine both handcraft and learnable templates for NFR classification. Table 2 shows the results for the 4 handcraft templates and 4 combined templates. From the table, we can find that adding soft templates can stabilize the performance of the handcraft templates. For example, the standard deviation of F1 score for the 4 combined templates (P11-P14) is 0.74 which is less than the four handcraft templates (1.00). Such finding provides the guidance that adding learnable tokens into handcraft templates can reduce the risk of randomness of handcraft templates for other prompt-based NFR classification techniques.

Table 1. Effectiveness of different handcraft templates and learnable templates. (P1-P4: handcraft templates. P5-P10: learnable templates)

Template	Precision	Recall	F1 score
P1	**83.59**%	**83.46**%	**83.52**%
P2	82.37%	82.50%	82.43%
P3	81.27%	81.97%	81.61%
P4	80.35%	81.27%	80.81%
P5	77.26%	76.64%	76.95%
P6	78.43%	79.17%	78.79%
P7	76.40%	78.35%	77.36%
P8	78.38%	78.12%	78.25%
P9	76.54%	76.53%	76.53%
P10	78.51%	77.60%	78.05%

Table 2. Effectiveness of combination of handcraft templates and learnable templates. (P1-P4: handcraft templates. P11-P14: combined templates)

Template	Precision	Recall	F1 score
P1	83.59%	83.46%	**83.52**%
P2	82.37%	82.50%	82.43%
P3	81.27%	81.97%	81.61%
P4	80.35%	81.27%	80.81%
P11	82.98%	**83.91**%	83.44%
P12	**83.69**%	82.76%	83.22%
P13	81.84%	82.03%	81.93%
P14	81.55%	82.07%	81.81%

6 Threats to Validity

The main external threat to the validity is the dataset we used. In our study, we use the widely used data PROMISE for NFR classification. But the labeling process of the data may not be accurate, leading to the model misinterpreting the words from the beginning.

7 Conclusion

In this paper, we conducted a comprehensive study to evaluate the performance of prompt-based non-functional requirements classification by designing various handcraft templates and soft templates on pre-trained model. Our experimental results show that handcraft templates can achieve best effectiveness (e.g., 83.52% in terms of F1 score) but with unstable performance for different templates. Also, the performance can become stable after learnable templates are inserted with handcraft templates.

References

1. Hugging Face. https://huggingface.co/. 2024
2. NLTK toolkit. https://www.nltk.org/. 2024
3. Sayyad, SJ. PROMISE software engineering repository. 2005
4. Binkhonain, Manal and Zhao, Liping. A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications: X, 2019
5. Amasaki, Sousuke and Leelaprute, Pattara. The effects of vectorization methods on non-functional requirements classification. 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)
6. EzzatiKarami, Mahtab and Madhavji, Nazim H, Automatically classifying non-functional requirements with feature extraction and supervised machine learning techniques: A research preview. Requirements Engineering: Foundation for Software Quality: 27th International Working Conference, REFSQ 2021, Essen, Germany, April 12–15, 2021.
7. Zhou, Ce and Li, Qian and Li, Chen and Yu, Jun and Liu, Yixin and Wang, Guangjing and Zhang, Kai and Ji, Cheng and Yan, Qiben and He, Lifang and others. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. arXiv preprint arXiv:2302.09419.
8. Devlin, Jacob. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
9. Qiu, Xipeng and Sun, Tianxiang and Xu, Yige and Shao, Yunfan and Dai, Ning and Huang, Xuanjing. Pre-trained models for natural language processing: A survey. Science China technological sciences, 2020
10. Liu, Yang and Zhang, Yao and Wang, Yixin and Hou, Feng and Yuan, Jin and Tian, Jiang and Zhang, Yang and Shi, Zhongchao and Fan, Jianping and He, Zhiqiang. A survey of visual transformers. IEEE Transactions on Neural Networks and Learning Systems, 2023
11. Brown, Tom B. Language models are few-shot learners. arXiv preprint arXiv:2005.14165. 2020
12. Liu, Xiao and Zheng, Yanan and Du, Zhengxiao and Ding, Ming and Qian, Yujie and Yang, Zhilin and Tang, Jie. GPT understands, too. AI Open, 2023
13. Schick, Timo and Schütze, Hinrich, Exploiting cloze questions for few shot text classification and natural language inference. arXiv preprint arXiv:2001.07676. 2020

14. Luo, Xianchang and Xue, Yinxing and Xing, Zhenchang and Sun, Jiamou. Prcbert: Prompt learning for requirement classification using bert-based pretrained language models. Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, 2022
15. Sahoo, Pranab and Singh, Ayush Kumar and Saha, Sriparna and Jain, Vinija and Mondal, Samrat and Chadha, Aman. A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv:2402.07927,2024
16. Abad, Zahra Shakeri Hossein and Karras, Oliver and Ghazi, Parisa and Glinz, Martin and Ruhe, Guenther and Schneider, Kurt. What works better? a study of classifying requirements. 2017 IEEE 25th International Requirements Engineering Conference (RE)
17. Navarro-Almanza, Raul and Juarez-Ramirez, Reyes and Licea, Guillermo. Towards supporting software engineering using deep learning: A case of software requirements classification. 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)
18. Dekhtyar, Alex and Fong, Vivian. Re data challenge: Requirements identification with word2vec and tensorflow. 2017 IEEE 25th International Requirements Engineering Conference (RE)
19. Rahimi, Nouf and Eassa, Fathy and Elrefaei, Lamiaa. One-and two-phase software requirement classification using ensemble deep learning. Entropy, 2021
20. Hey, Tobias and Keim, Jan and Koziolek, Anne and Tichy, Walter F. Norbert: Transfer learning for requirements classification. 2020 IEEE 28th international requirements engineering conference (RE)
21. Chatterjee, Ranit and Ahmed, Abdul and Anish, Preethu Rose and Suman, Brijendra and Lawhatre, Prashant and Ghaisas, Smita. A pipeline for automating labeling to prediction in classification of nfrs. 2021 IEEE 29th International Requirements Engineering Conference (RE)
22. Loshchilov, I. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101. 2017