

Algorithm for Improving Software Requirements Analysis Using Natural Language Processing Technology

Ping Xu^{1*}

¹ School of Information Science and Technology, Taishan University, Taian, China

Email Address

taianxp@163.com (Ping Xu)

*Correspondence: taianxp@163.com

Received: 3 October 2025; **Accepted:** 12 November 2025; **Published:** 30 November 2025

Abstract:

With the increasing complexity of software system, the accuracy and efficiency of software requirements analysis become the key. This article focuses on improving the software requirements analysis algorithm by using NLP (Natural Language Processing) technology. In this article, an algorithm based on NLP is designed in detail, which covers the preprocessing of demand text, removing redundancy, correcting errors and unifying the format. Feature extraction is carried out by using word vector model, and key information is captured by combining part-of-speech tagging and named entity recognition. At the same time, this article completes the requirements classification with the help of CNN (Convolutional Neural Network) in DL (Deep Learning) model. Identify demand relationship by syntactic analysis and semantic role labeling. In the experiment, data sets of various demand categories are constructed and compared with traditional rule-based and naive Bayesian algorithms. The results show that the accuracy, recall and F1 value of this algorithm are excellent in all kinds of requirements, and the accuracy, recall and F1 value of functional requirements are 0.92, 0.90 and 0.91 respectively. And with the increase of the training set, the performance of the algorithm is improved, and the performance is better when the scale reaches 10000 samples. It can provide an effective way to improve the quality of software requirements analysis.

Keywords:

NLP Technology, Software Requirements Analysis, Algorithm Design, Experimental Evaluation, Deep Learning

1. Introduction

In today's digital age, the complexity and scale of software systems are rising continuously. As the beginning and key link of software development, software requirements analysis is self-evident [1]. Software requirements analysis aims to accurately capture users' expectations of software functions, performance and other characteristics, and lay a solid foundation for subsequent software design, development and testing [2]. Accurate software requirements analysis can

significantly reduce changes and errors in software development, reduce costs and improve software quality. On the other hand, if there is deviation in requirements analysis, it is very likely that the software project will fail [3]. However, the traditional software requirements analysis method faces many challenges. On the one hand, the requirement acquisition stage often relies on manual communication and document arrangement, which is not only inefficient, but also prone to information omission or misunderstanding [4]. On the other hand, requirements documents are mostly described in natural language, and its fuzziness and ambiguity make it difficult to understand and analyze requirements accurately [5]. Therefore, exploring more effective software requirements analysis techniques and methods has become a research hotspot in the field of software engineering.

As an interdisciplinary field of computer science and linguistics, NLP technology focuses on making computers understand, process and generate human natural language [6]. In recent years, with the rapid development of DL and other technologies, NLP has made breakthroughs in many fields, such as text classification, information extraction, machine translation and so on. Introducing NLP technology into software requirements analysis is expected to solve the problems faced by traditional methods and realize automatic and intelligent analysis of software requirements [7]. By using NLP technology, requirements documents can be efficiently analyzed in morphology, syntax and semantics, key information can be accurately extracted, and requirements patterns and relationships can be identified, thus improving the accuracy and efficiency of requirements analysis [8]. At the same time, NLP technology can also help the automatic classification and prioritization of requirements, and provide strong support for software project management. This study focuses on improving the algorithm of software requirements analysis by using NLP technology, and deeply discusses the application potential and realization path of NLP technology in software requirements analysis. The purpose of this study is to propose an efficient and accurate software requirements analysis algorithm, which will contribute to improving software quality and development efficiency.

2. Software Requirement Analysis and the Theoretical Origin of NLP Technology

In the early days, software development mainly focused on programming. With the increase of software scale and complexity, people gradually realized the importance of clearly defining requirements in the early stage of development [9]. Starting from structured method, it emphasizes top-down and gradual refinement, and describes system functions and data flow through tools such as data flow diagram. After that, the object-oriented analysis method appeared, taking the object as the core, and focusing the requirement analysis on the object and its interaction.

The theoretical source of NLP technology involves linguistics, computer science and other disciplines. Linguistics provides a theoretical basis for it, such as language structure and grammatical rules, so that researchers can understand the composition and laws of human language. The knowledge of algorithm design and data structure in computer science provides technical means for the realization of NLP technology. From the early rule-based method, processing natural language by writing grammar rules manually, to the statistical method, using large-scale corpus for probability calculation to realize language processing, and then to the booming DL method, NLP technology has been continuously innovated [10]. Its theoretical development always revolves around how to make computers better understand and process human natural

language, which is in line with the processing requirements of natural language requirements documents in software requirements analysis, laying a theoretical foundation for the combination of the two.

3. Design of Software Requirements Analysis Algorithm Based on NLP

The algorithm based on NLP mainly covers the key links such as requirements text preprocessing, feature extraction, requirements classification and relationship identification. In the pre-processing stage of requirements text, the original software requirements document usually contains problems such as nonstandard format, redundant information and spelling errors. In order to transform the text into a form suitable for subsequent analysis, a series of operations are needed. First, clean up the text, and remove irrelevant punctuation marks, special characters and stop words. Let the original text be T , the text after stop words removal be T' , and the stop words set be S , then the operation can be expressed as:

$$T' = T - S \quad (1)$$

Although these stop words are indispensable in natural language expression, they make little contribution to the essence of demand analysis.

For abbreviations and abbreviated forms, restore and expand to ensure the integrity of text information. Assuming that the abbreviation set is A , and its corresponding complete form set is B , the abbreviation replacement operation in the text can be expressed as follows: for $a_i \in A$, if there is $b_i \in B$ corresponding to it, replace the a_i in the text T' with b_i to get a new text T'' .

In view of the spelling mistakes in the text, the spelling checker is used to correct them. If the spelling error set is E and the correct form set is C , the spelling error correction operation can be expressed as follows: for $e_i \in E$, if there is $c_i \in C$ corresponding to it, the e_i in the text T'' will be replaced by c_i , and finally high-quality text data T_{final} will be provided for subsequent analysis.

Feature extraction is very important, and its purpose is to extract features that can represent the core content of software requirements from the preprocessed text T_{final} . In this algorithm, the word vector model Word2Vec is used to map each word to a low-dimensional vector space, so as to capture the semantic relationship between words. Assuming that the vocabulary is V , for the word $w_i \in V$, its word vector representation $\vec{v}_{w_i}$ is obtained through training, and its dimension is d , namely:

$$\vec{v}_{w_i} \in \mathbb{R}^d \quad (2)$$

These vectors not only contain the semantic information of words, but also reflect their context. By vectorizing the words in the demand text, the feature vector representation of the text is formed. Assuming that the text is composed of word sequence $w_1, w_2, \dots, w_n$, the feature vector $\vec{V}$ of the text can be obtained by some aggregation method, simple average method:

$$\vec{V} = \frac{1}{n} \sum_{i=1}^n \vec{v}_{w_i} \quad (3)$$

Combined with part-of-speech tagging and named entity recognition technology, key nouns, verbs and specific entity names in the text are extracted. These key information can more accurately reflect the core content of software requirements.

Requirements classification is one of the core functions of the algorithm, which aims to classify software requirements according to different categories, so as to facilitate the project team to deal with them in a targeted manner. This study adopts the classification model CNN based on DL. Assuming that the input text feature vector is $\vec{V}$, after a series of operations such as convolution layer and pooling layer, the feature representation $\vec{H}$ is obtained. Let the convolution kernel set be K , and the convolution operation can be expressed as:

$$\vec{H} = f(\sum_{k_j \in K} \vec{V} * k_j + b) \quad (4)$$

Where f is the activation function, $*$ is the convolution operation, and b is the offset term.

When training the classification model, a large number of labeled software requirements texts are collected as training data, covering common categories such as functional requirements, performance requirements and security requirements. By learning these training data, the model constructs the mapping relationship between text features and demand categories. Assuming that the category set is C , the prediction function of the model is:

$$P(\vec{V}, \theta) \quad (5)$$

Where θ is the model parameter, the model is trained by minimizing the loss function L , and the cross entropy loss function:

$$L = - \sum_{c_i \in C} y_{c_i} \log (P(\vec{V}, \theta)_{c_i}) \quad (6)$$

Where y_{c_i} is the real label.

There are often various complicated relationships between software requirements, such as dependency, causality and so on. In order to identify these relationships, the algorithm uses syntactic analysis technology to construct the syntactic tree of the required text. Let the sentence S consist of word sequence $w_1, w_2, \dots, w_n$, and get the syntax tree T_{syntax} through syntax analysis. The construction process of the syntax tree can be realized through syntax analysis algorithm. Dependency parsing algorithm, the core of which is to determine the dependency between each word and other words, assumes that the set of dependencies is D , and for each word w_i , there exists the dependency of $d_{ij} \in D$ representing w_i and w_j . By analyzing these dependencies, the potential logical connection between requirements can be mined. Combined with semantic role labeling technology, the semantic role of each word in the sentence is determined, and the semantic association between requirements is further clarified.

4. Algorithm Experiment and Evaluation

In this study, a series of experiments were carefully designed and carried out, and the experimental results were quantitatively analyzed by scientific and reasonable evaluation indicators. A large number of software requirements documents were collected from several actual software projects, and after screening and sorting, experimental data sets containing different types of requirements were constructed. The data set covers functional requirements, performance requirements, security requirements, user interface requirements and other categories to ensure that the diversity of software requirements can be fully reflected. In order to highlight the advantages of this algorithm, the rule-based software requirements analysis algorithm

and the naive Bayesian classification algorithm based on machine learning are selected as the comparison objects. Traditional rule-based algorithms rely on manually written rules for requirements analysis, while Naive Bayesian algorithm is a classic machine learning classification method, which is widely used in the field of text classification.

Accuracy, recall and F1 value were used as the main evaluation indexes. This algorithm, rule-based algorithm and naive Bayes algorithm are tested on the test set respectively, and the following results are obtained (see Figure 1):

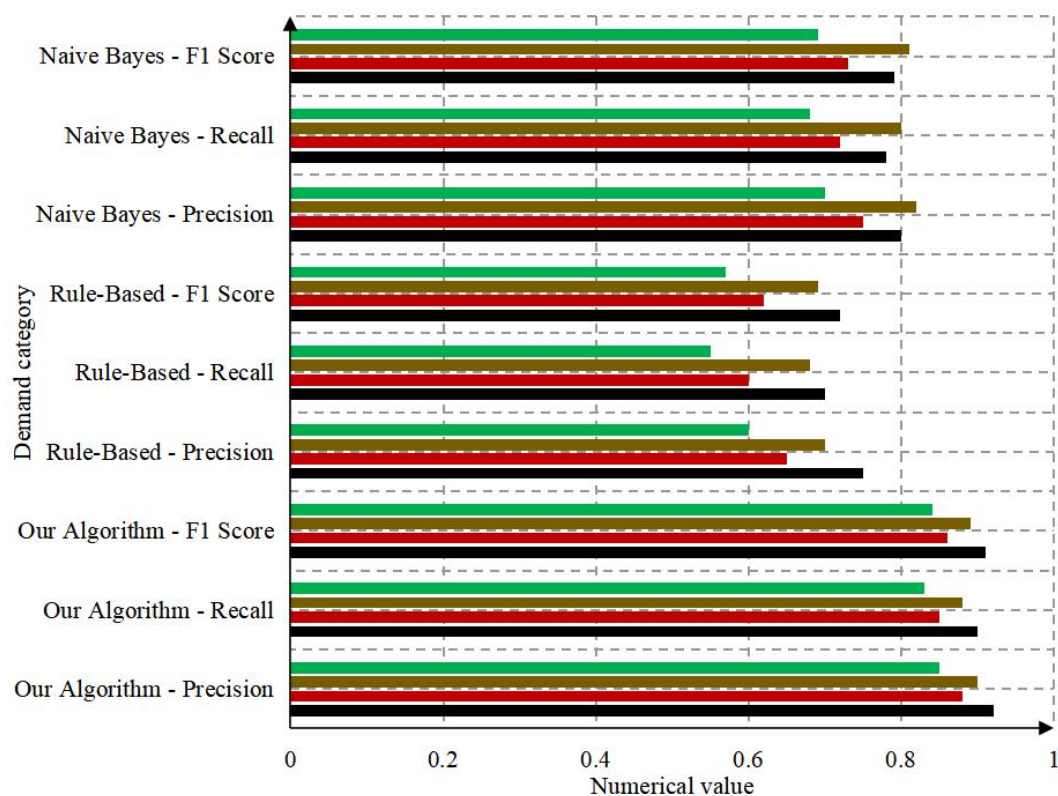


Figure 1. Performance comparison of different algorithms in various requirements.

As can be seen from Figure 1, the accuracy, recall and F1 value of this algorithm are higher than those of rule-based algorithm and naive Bayesian algorithm in various requirements. This shows that this algorithm has better performance in identifying different types of software requirements and can classify software requirements more accurately.

In order to explore the influence of data set size on the performance of this algorithm, the algorithm is trained with training sets of different sizes and tested on the same test set. The results are as follows (see Figure 2):

From the analysis of Figure 2, it can be seen that with the increase of the training set size, the performance of this algorithm is gradually improved. This shows that this algorithm has good adaptability to data, and more training data can help the algorithm learn more features, thus improving its classification performance. When the training set reaches 10000 samples, the algorithm achieves high accuracy, recall and F1 value (above 0.95) on the test set, which further verifies the effectiveness and stability of the algorithm in practical application.

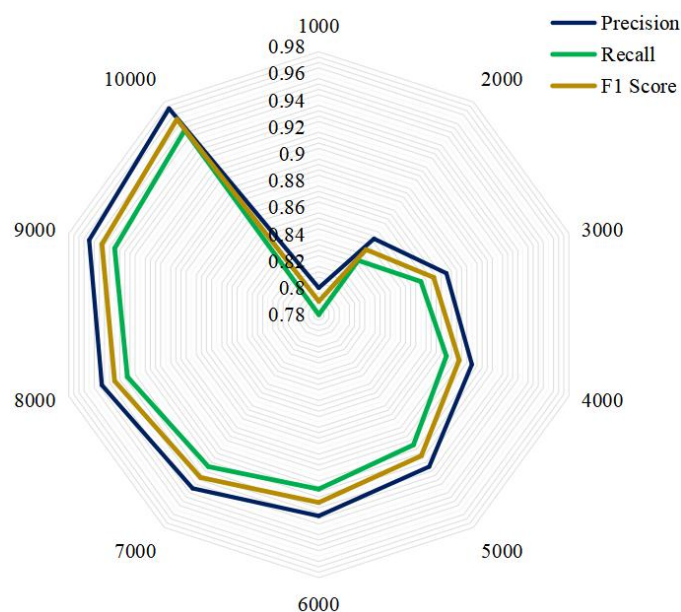


Figure 2. Performance of this algorithm under different scale data sets.

5. Conclusions

In software development, software requirements analysis is very important, but traditional methods face many difficulties. The development of NLP technology brings opportunities to solve these problems. In this article, the software requirements analysis algorithm based on NLP is carefully designed. Through pretreatment, the original requirements documents are cleaned and standardized, which lays a good foundation for subsequent analysis; Accurate feature extraction is realized by word vector model, part-of-speech tagging and named entity recognition. Using DL classification model to complete efficient demand classification; Using syntactic analysis and semantic role labeling to identify complex relationships between requirements. In the experimental evaluation, a variety of data sets are constructed and compared with the traditional rule-based algorithm and naive Bayes algorithm. The experimental results fully demonstrate the excellent performance of this algorithm. In all kinds of requirements classification tasks, the accuracy, recall and F1 value of this algorithm far exceed those of the contrast algorithm. In terms of functional requirements, the accuracy rate is 0.92, the recall rate is 0.90, and the F1 value is 0.91, which means that the algorithm can identify functional requirements more accurately and reduce misjudgments and missed judgments. Tests on data sets of different sizes show that the performance of the algorithm is steadily improved with the increase of the training set size. When the training set size reaches 10,000 samples, the accuracy, recall and F1 value of the algorithm all reach a high level (above 0.95). This reflects the good adaptability of the algorithm to data and the stability and reliability in practical application. In this study, NLP technology is successfully used to improve the software requirements analysis algorithm, which greatly improves the accuracy and efficiency of software requirements analysis.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

References

- [1] Li, X.J.; Yang, Z.B.; Wang, H.F. et al. A Method for Generating Verification Properties Oriented to Natural Language Requirements. *Journal of Chinese Computer Systems*, 2024, 45(01), 84-92. DOI: 10.20009/j.cnki.21-1106/TP.2022-0328.
- [2] Peng, Z. A Requirement Change Impact Analysis Method Based on Natural Language Description. *Digital Technology and Application*, 2020, 38(03), 209-211. DOI: 10.19695/j.cnki.cn12-1369.2020.03.102.
- [3] Li, G.L.; Shen, G.H.; Huang, Z.Q. et al. Research on a Security Requirement Acquisition Method for Natural Language Requirements. *Journal of Chinese Computer Systems*, 2023, 44(12), 2646-2655. DOI: 10.20009/j.cnki.21-1106/TP.2022-0284.
- [4] Zhai, Y.K.; Wei, D.L.; Lu, W. et al. Identification and Analysis of Product Improvement Requirements Considering User Attention in Online Reviews—Based on the KANO-IPA Model. *Science and Technology Management Research*, 2023, 43(9), 95-104. DOI: 10.3969/j.issn.1000-7695.2023.9.013.
- [5] Qin, Q.Q.; Wei, Q.G. A Software Defect Prediction Method Combined with Software Development Process. *Application Research of Computers*, 2020, 37(S2), 210-213.
- [6] Zhao, Y.B.; Jiang, F.; Li, P.F. A Multi-level Coherent Text Segmentation Method Based on BERT. *Computer Applications and Software*, 2024, 41(10), 262-268. DOI: 10.3969/j.issn.1000-386x.2024.10.039.
- [7] Qin, N.; Ma, L.; Huang, R. Software Security Requirement Analysis and Verification Method Based on System Theoretic Process Analysis. *Journal of Computer Applications*, 2020, 40(11), 3261-3266. DOI: 10.11772/j.issn.1001-9081.2020040548.
- [8] Zeng, J.; Wang, Z.W.; Yu, Y. et al. A Survey of Word Embedding Methods in the Field of Natural Language Processing. *Journal of Frontiers of Computer Science and Technology*, 2024, 18(1), 24-43. DOI: 10.3778/j.issn.1673-9418.2303056.
- [9] Lan, F.; Zhang, Y. A Natural Language Causal Relation Extraction Method Based on Hybrid CNN. *Computer Applications and Software*, 2021, 38(10), 184-188+266. DOI: 10.3969/j.issn.1000-386x.2021.10.029.
- [10] Ding, Q. Design of Anti-malware Attack Test System Based on Protocol Analysis Technology. *Modern Electronics Technique*, 2020, 43(23), 54-57. DOI: 10.16652/j.issn.1004-373x.2020.23.012.



© 2025 by the author(s); licensee International Technology and Science Publications (ITS), this work for open access publication is under the Creative Commons Attribution International License (CC BY 4.0). (<http://creativecommons.org/licenses/by/4.0/>)