e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 5, Issue 8, August 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.54

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 5, Issue 8, August 2022 |

| DOI:10.15680/IJMRSET.2022.0508011|

Enhancing Software Requirements with Semantic Similarity Analysis

Madhura, Dr. Todmal Satish Ramchandra

Scholar, Department of Computer Science, Sunrise University, Alwar, India

Associate Professor, Department of Computer Science, Sunrise University, Alwar, India

ABSTRACT: The complexity of software development has increased significantly, necessitating precise and efficient analysis of software requirements. One of the primary challenges in software engineering is ensuring that requirements are clear, consistent, and aligned with stakeholder intentions. This paper introduces an approach to enhance software requirements analysis using semantic similarity techniques. By leveraging natural language processing (NLP) methods, this approach can automatically identify and analyze the semantic relationships between words and sentences within requirement documents. This not only helps in detecting ambiguities and inconsistencies but also facilitates better communication among stakeholders. The results from applying this method to various case studies demonstrate its potential to improve the quality and clarity of software requirements.

KEYWORDS: Semantic similarity, Software requirements, Requirements analysis, Natural language processing (NLP), Word embeddings.

I. INTRODUCTION

The complexity and scale of modern software systems have evolved significantly, leading to increasingly intricate requirements that must be accurately captured and analyzed. Software requirements are critical to the development process, serving as the blueprint for design, implementation, and testing. These requirements need to be clear, consistent, and comprehensive to ensure that the final software product aligns with stakeholders' needs and expectations. However, the process of gathering, documenting, and analyzing requirements is fraught with challenges. Requirements documents often suffer from ambiguities, inconsistencies, and misinterpretations, which can lead to costly errors and rework later in the development cycle. Traditional methods for requirements analysis rely heavily on manual review and expert judgment. While experienced requirements engineers can identify and address many issues, this process is time-consuming and prone to human error. Furthermore, as requirements documents grow in size and complexity, manual methods become increasingly inadequate. The need for automated tools and techniques to enhance the analysis of software requirements has become more pressing. This is where semantic similarity analysis comes into play. Semantic similarity analysis leverages techniques from natural language processing (NLP) to evaluate and compare the meaning of words and sentences within requirement documents. Unlike traditional keyword-based methods, semantic similarity focuses on the contextual and conceptual relationships between text elements. By understanding the underlying meaning of terms and statements, semantic similarity analysis can identify nuances that may be missed by superficial keyword matching. At the core of semantic similarity analysis are various NLP techniques, including word embeddings and sentence embeddings. Word embeddings, such as those generated by Word2Vec, GloVe, or FastText, represent words as dense vectors in a high-dimensional space. These vectors capture the semantic relationships between words based on their usage in large text corpora. Sentence embeddings extend this concept to entire sentences, providing a way to represent and compare the meanings of longer text segments. Advanced models like BERT and Sentence-BERT further enhance this capability by incorporating contextual information, enabling a deeper understanding of the text.

Applying semantic similarity analysis to software requirements offers several potential benefits. Firstly, it can help detect ambiguities by identifying semantically similar but differently phrased requirements. For instance, two requirements that describe the same functionality using different terminology can be flagged as potential duplicates or inconsistencies. This helps ensure that all aspects of the software's functionality are clearly defined and avoids redundant or conflicting requirements.

Secondly, semantic similarity analysis can enhance the consistency of terminology throughout the requirements document. In large projects, different stakeholders may use varying terms to refer to the same concept. Semantic similarity analysis can identify these variations and suggest standardization, improving the overall clarity and coherence

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 5, Issue 8, August 2022 |

| DOI:10.15680/IJMRSET.2022.0508011|

of the requirements. This is particularly valuable in complex projects where multiple teams are involved and consistent terminology is crucial for effective communication.

Moreover, semantic similarity analysis can aid in the validation and verification of requirements. By comparing the semantic content of requirements with established standards or existing documentation, it is possible to identify discrepancies and ensure that the requirements meet predefined criteria. This can help in ensuring that all necessary aspects of the software are addressed and that the requirements align with stakeholder expectations.

Despite its advantages, the application of semantic similarity analysis in software requirements engineering is not without challenges. One significant challenge is the quality and accuracy of the NLP models used for semantic analysis. While models like BERT have demonstrated impressive performance in various NLP tasks, they are not infallible. The effectiveness of semantic similarity analysis depends on the quality of the embeddings and the ability of the models to capture the specific nuances of domain-specific language. Therefore, it is essential to carefully select and fine-tune these models to suit the context of software requirements.

Another challenge is integrating semantic similarity analysis with existing requirements engineering processes. While automated analysis can provide valuable insights, it should complement rather than replace traditional methods The results from semantic similarity analysis need to be interpreted and validated by human experts to ensure that they are meaningful and actionable. Therefore, a hybrid approach that combines.

automated analysis with expert review may be the most effective way to enhance the quality of software requirements.
In enhancing software requirements with semantic similarity analysis represents a promising advancement in requirements engineering. By leveraging NLP techniques to understand and compare the meaning of text elements, this approach can address common issues such as ambiguity, inconsistency, and terminology variation. The potential benefits include improved clarity, consistency, and alignment with stakeholder needs, which can ultimately lead to more successful software projects. However, careful consideration of the challenges and integration with traditional methods is essential for realizing the full potential of semantic similarity analysis in this domain.

II. METHODOLOGY

Our proposed approach for enhancing software requirements with semantic similarity analysis consists of the following steps:

1. **Preprocessing**: Requirement documents are first preprocessed to remove noise and standardize the text. This includes steps such as tokenization, lemmatization, and stop-word removal.
2. **Word-Level Similarity**: We compute word-level semantic similarity using word embeddings, which represent words as continuous vectors in a high-dimensional space. Popular word embedding techniques include Word2Vec, GloVe, and FastText. These embeddings allow us to measure the cosine similarity between words to assess their semantic relatedness.
3. **Sentence-Level Similarity**: To capture the meaning of entire sentences, we use sentence embeddings, which extend word embeddings to represent whole sentences. Methods such as Sentence-BERT or transformer-based models like BERT and GPT can be used to generate these embeddings. Sentence-level similarity is then computed using cosine similarity or other distance metrics.
4. **Analysis and Evaluation**: The computed similarities are used to analyze the requirement documents. We look for semantically similar or contradictory statements, duplicate requirements, and inconsistent terminology. This analysis helps in identifying potential issues and areas for improvement.
5. **Feedback and Refinement**: Based on the analysis, feedback is provided to stakeholders, suggesting revisions to improve the clarity and consistency of the requirements. The process can be iterated to refine the documents further.

III. CASE STUDY AND RESULTS

To evaluate the effectiveness of our approach, we conducted a case study using a set of software requirements from a real-world project. The requirements were analyzed using our semantic similarity analysis method, and the results were compared to a manual analysis conducted by experienced requirements engineers.

Our method successfully identified several duplicate and contradictory requirements that were overlooked in the manual analysis. Additionally, it highlighted inconsistencies in terminology and suggested areas where the requirements could

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.54| Monthly, Peer Reviewed & Referred Journal|

| Volume 5, Issue 8, August 2022 |

| DOI:10.15680/IJMRSET.2022.0508011|

be made clearer. Overall, the use of semantic similarity analysis significantly improved the quality of the requirements document, reducing ambiguities and enhancing stakeholder understanding.

The results of our case study demonstrate the potential of semantic similarity analysis in enhancing software requirements. By automating the process of analyzing semantic content, our approach can save time and reduce the likelihood of errors in requirements documents. It also facilitates better communication among stakeholders by ensuring that requirements are clear, consistent, and aligned with stakeholder intentions
.

However, there are some limitations to our approach. The accuracy of semantic similarity analysis depends on the quality of the embeddings and models used. In some cases, the models may not fully capture the nuances of domain-specific language, leading to incorrect assessments. Additionally, the approach may require customization to handle specific project contexts or types of requirements.

IV. CONCLUSION

Semantic similarity analysis is a powerful tool for enhancing software requirements engineering. By leveraging NLP techniques to assess the semantic content of requirement documents, we can improve their clarity, consistency, and overall quality. Our proposed methodology has shown promising results in a real-world case study, demonstrating its potential to streamline the requirements analysis process and reduce ambiguities. Future work will focus on refining the approach, incorporating domain-specific knowledge, and extending its application to other areas of software engineering. Future research could explore several areas to further enhance the effectiveness of semantic similarity analysis in software requirements engineering. One direction is to incorporate domain-specific ontologies and knowledge bases to improve the accuracy of semantic analysis. Another area of interest is the development of more sophisticated models that can capture the context and intent behind requirements more effectively. Additionally, integrating this approach with other requirements engineering tools and techniques could provide a more comprehensive solution for managing software requirements.

REFERENCES

1. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. 2013;26:3111-9.
2. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. Proc 2014 Conf Empir Methods Nat Lang Process (EMNLP). 2014;1532-43.
3. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. Proc 2019 Conf North Am Chapter Assoc Comput Linguist Human Lang Technol. 2019;4171-86.
4. Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proc 2019 Conf Empir Methods Nat Lang Process (EMNLP). 2019;3980-90.
5. Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. Proc 2015 Conf Empir Methods Nat Lang Process (EMNLP). 2015;1412-21.
6. Yang Z, Dai Z, Yang Y, Carbonell JG, Salakhutdinov R, Xie J. XLNet: Generalized autoregressive pretraining for language understanding. Proc 2019 Conf Empir Methods Nat Lang Process (EMNLP). 2019;5753-63.
7. Kumar A, Vassilev V. Analyzing software requirements using text mining techniques: A systematic review. J Softw Evol Process. 2020;32(5).
8. Nair R, Yadav S. Automated detection of requirement ambiguity using NLP techniques: A comparative study. Softw Pract Exp. 2021;51(7):1425-41.
9. Gómez-Adorno H, García-Serrano A. Semantic similarity measures in natural language processing: A review. J Comput Linguist Intell Text Process. 2018;29(4):123-46.
10. Zhang Y, Wang H. Leveraging transformer-based models for improved requirements analysis in software engineering. IEEE Trans Softw Eng. 2021;47(3):1187-201.

INNO SPACE

SJIF Scientific Journal Impact Factor

Impact Factor

7.54

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com