International Journal of Cloud Applications and Computing
Volume 14 • Issue 1

Detecting Ambiguities in Requirement Documents Written in Arabic Using Machine Learning Algorithms

Ahmad Althunibat, Al-Zaytoonah University of Jordan, Jordan
Bayan Alsawareah, Al-Zaytoonah University of Jordan, Jordan
Siti Sarah Maidin, INTI International University, Malaysia
https://orcid.org/0000-0003-0714-2186
Belal Hawashin, Al-Zaytoonah University of Jordan, Jordan
Iqbal Jebril, Al-Zaytoonah University of Jordan, Jordan
Belal Zaqaibeh, Jadara University, Jordan
Haneen A. Al-khawaja, Applied Science Research Center, Applied Science Private University, Jordan*
https://orcid.org/0000-0003-4607-9394

ABSTRACT

The identification of ambiguities in Arabic requirement documents plays a crucial role in requirements engineering. This is because the quality of requirements directly impacts the overall success of software development projects. Traditionally, engineers have used manual methods to evaluate requirement quality, leading to a time-consuming and subjective process that is prone to errors. This study explores the use of machine learning algorithms to automate the assessment of requirements expressed in natural language. The study aims to compare various machine learning algorithms according to their abilities in classifying requirements written in Arabic as decision tree. The findings reveal that random forest outperformed all stemmers, achieving an accuracy of 0.95 without employing a stemmer, 0.99 with the ISRI stemmer, and 0.97 with the Arabic light stemmer. These results highlight the robustness and practicality of the random forest algorithm.

KEYWORDS

Arabic Requirements Ambiguous Requirement Classification, Machine Learning AraBERT, Software Engineering, Unambiguous Requirements

Software engineering is the application of engineering principles to ensure the quality, reliability, and efficiency of software systems. The Institute of Electrical and Electronics Engineers (IEEE) defines software engineering as "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" (IEEE, 1990). As technology advances, software engineering faces challenges such as the classification and ambiguity of software system requirements. Requirements are fundamental to software development, and their ambiguity can lead to misunderstandings, miscommunications, and project failure. Understanding and addressing these

DOI: 10.4018/IJCAC.339563
*Corresponding Author

challenges is crucial, and model-driven engineering approaches can enhance requirement clarity and consistency.

A significant obstacle for academics and developers working on Arabic natural language processing (NLP) applications is the scarcity of Arabic datasets, particularly in recognizing ambiguous requirements. Failure to detect and resolve ambiguity early in the software requirements specification (SRS) can result in various defects with serious consequences for the software development process. Ambiguities in software requirements, defined as sentences with multiple meanings, can lead to faults and compromise software reliability, especially in safety-critical systems.

This research addresses the challenge of detecting ambiguous requirements in Arabic language software development by proposing a novel approach that employs supervised learning algorithms for automated classification. This approach contributes significantly to Arabic NLP, offering a potential solution to improve software development accuracy in Arabic-speaking regions. By providing a robust means of detecting ambiguous requirements, the proposed method has the potential to reduce risks associated with Arabic software development, ultimately leading to higher-quality software products.

The importance of an efficient method for classifying Arabic requirements as ambiguous or non-ambiguous is twofold. First, in the research domain, as there are limited prior studies in this area, this method serves as a pioneering effort that could stimulate further research. Second, from a practical standpoint, various stakeholders stand to benefit from improved requirements engineering processes.

The key contributions of this work include proposing the use of machine learning (ML) to classify ambiguous requirements in Arabic, comparing several classifiers to identify the most effective method for this task, and creating a dataset of Arabic requirements for future research. The subsequent sections encompass a literature review, methodology, experimental work, and conclusions, providing a comprehensive exploration of the proposed approach.

RELATED WORK

In the realm of software development, understanding stakeholder needs is crucial for designing complex software systems (Althunibat et al., 2022). Stakeholders, often users, contribute NLP-written requirements for large-scale projects. Ko et al. (2007) proposed an approach wherein initial data needs are automatically categorized into topics, reflecting political analyst perspectives. Experiments, utilizing datasets in both Korean and English, validate the efficacy of this strategy. This highlights the potential for an internet-based requirements analysis-supporting system to efficiently gather and evaluate dispersed end-user requirements via the network.

Moving forward, support vector machine (SVM) algorithms have garnered attention for their ideal academic characteristics and high performance (Al Qaisi et al., 2021). Yang et al. (2010) delved into the analysis of support vector characteristics, presenting a novel learning process that incorporates SVM classification algorithms. The algorithm, rooted in the equivalence of classification between support vector sets, employs incremental learning to accumulate data. Experimental results indicate its potential to expedite training processes, reduce storage costs, and maintain organizational accuracy (Quba et al., 2021).

Artificial intelligence (AI) and deep learning (DL) come to the forefront in the work of Navarro-Almanza et al. (2017). They recommend using a convolutional neural network (CNN) model to categorize software requirements, showcasing promising results on the PROMISE corpus dataset. This dataset, with pre-grouped and labeled criteria for both functional requirements (FR) and non-functional requirements (NFR), serves as a valuable resource for evaluating the suggested model. (Gill et al., 2014)

Lu and Liang (2017) further contributed to understanding user requirements by breaking them down into FRs and NFRs, including usability, portability, performance, and reliability. Their research involved diverse methods such as bag of words (BoW), CHI2, TF-IDF, and AUR-BoW, as well as ML algorithms like J48, naive Bayes, and bagging. Comparative analysis reveals that the bagging