Validating Requirement Specification using Text Mining Analysis

1st Muhamad Azmi Rizkifar
Department of Informatics
Telkom University
Bandung, Indonesia
azmirf@student.telkomuniversity.ac.id

2nd Yudi Priyadi
Department of Software Engineering
Telkom University
Bandung, Indonesia
whyphi@telkomuniversity.ac.id

3rd Muhammad Johan Alibasa
Department of Software Engineering
Telkom University
Bandung, Indonesia
alibasa@telkomuniversity.ac.id

2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS) | 979-8-3503-2776-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICE3IS59323.2023.10335293

***Abstract*—Requirement elicitation is the first step of requirement engineering where software developers focus on obtaining the users' needs and produce a requirement specification. This activity can be accomplished through conducting interviews. However, the requirement specified might not conform to the actual interview transcripts from the elicitation process due to perspective differences between the clients and developers. This study aims to propose a method to validate the requirement specification using text mining analysis. The method involves text preprocessing and text similarity analysis between the specification and elicitation transcripts. The proposed method is validated using interview elicitation transcripts and the software requirement specification (SRS) from the Baker's Corner application. The results demonstrate that the proposed method can effectively validate requirement specifications, producing comparable results to the manual validation process. This finding highlights the potential of automating the requirement validation process using text mining techniques. Additionally, the paper offers recommendations for future studies and developers to enhance the generation of software requirements.**

***Keywords*—Requirement Elicitation, Requirement Specification, Functional Requirement, Non-Functional Requirement, Text Mining.**

I. Introduction

The initial and one the most crucial stage in developing software is requirement engineering. The output of this stage is Software Requirement Specification (SRS) [1]. This document describes all system requirements that must be met by the developed software [2]. Several processes including identification and collection of requirements are required to design artifacts in the SRS. These processes are categorized into Requirement Elicitation, that can be done by performing direct communications such as focus group and interviews, or indirect communication through online questionnaires to parties interested in the project [3], [4]. During requirement elicitation, natural language is might be used as the obtained results can be from discussions and interviews, confirming the applicability of Natural Language Processing (NLP) for the data analysis [5].

Requirement specification consists of requirements that a system must meet or fulfil, contracts and conformance standards based on user needs and expectations. Therefore, it is highly recommended for a system analyst or developer to have the same understanding and perspective on requirement specifications so that the software development results follow user needs [6]. In several instances, there are differences in perspective between clients and developers related to software requirements specifications, leading to incorrect requirement specifications. Therefore, a study is required to help validate the software requirements. As both requirement specification and elicitation can use NLP, there is an opportunity of using text analysis or text mining method to automate the validation process.

This research aims to propose a method to validate the requirements provided in the SRS document using text mining analysis. The validation method utilizes the interview transcripts from the requirement elicitation process as inputs. The method then compares the similarity and suitability of the transcripts with the provided functional and non-functional requirements included in the SRS document. Our method also describes the text pre-processing methods prior to text analysis. To evaluate our method, this research uses the SRS documents from an application called Baker's Corner. Baker's Corner is an Android-based pastry product marketing and sales application that is available in Google Play. The selection of the Baker's Corner application was based on the ease of accessing direct interviews with stakeholders and it being an application previously developed by the author. Therefore, the SRS documents of the Baker's Corner application are used as a case study and can be beneficial for general application development purposes. The research aims to mitigate discrepancies in user and developer perceptions that can lead to requirement misalignment. The contributions of our study are as follows:

1. Proposing a method to validate functional and non-functional requirements using text mining analysis, including pre-processing and text similarity, by comparing them against the requirement elicitation artefacts.
2. Evaluating our proposed method comprehensively by performing reliability testing with Cohen's Kappa.
3. Generate a similarity value from the text mining analysis process to compare it with the results of reliability testing for validating the requirement specification.
4. Providing discussions related to factors that might affect the low similarity score and some software requirement improvement suggestions.

II. Related Work

A. Requirement Elicitation and Specifications

Elicitation is an initiation process and method of collecting information that will be analyzed, designed, and tested to obtain a design that complies with the software requirements specification [4]. Several categories of requirements elicitation techniques are grouped into four parts: discussion methods, such as interviews. Observational methods, such as ethnography. Analytical methods, such as background reading, and synthetic methods, such as prototyping [7]. The initial stage in conducting the needs analysis process is elicitation. In conducting elicitation, it is necessary to collect information which will then be analyzed, modeled, and validated to obtain data conformity between user needs and elicitation results [8]. The Requirement Elicitation stage is essential in determining the quality of the software to be built. A complete requirement design can result in cost overruns and more development efforts [9], [10].

SRS is a document that contains a set of descriptive information of complete software requirements with various types of system modelling made to support the creation and development of software [5], [11]. SRS must be created because it can define all stakeholder needs and system design. Software development can be more structured through SRS because it generally provides direction through a road map to all parties involved to ensure the results meet user needs [5].

Generally, Software Requirement Specification (SRS) artifacts focus on Functional Requirements. In most SRS documents, Non-Functional Requirement is only sometimes formally defined throughout its content, and Non-Functional Requirement is often merged with Functional Requirement [8], [12].

To determine the system output expected by users, Functional Requirement is used when users provide specific inputs if they meet certain conditions [13]. Generally, the primary material in research related to Requirement Statements is Functional Requirement because Functional Requirement is the direct reference in creating various types of features per user expectations. However, in some cases, the output of the Functional Requirement is often criticized by users because they think the results are very different from what they expected [4], [8], [14].

B. Text Pre-Processing and Text Mining

Text Mining or can be called Text Data Mining is the activity of exploring and analyzing knowledge from textual databases to obtain information from a set of data in the form of text, which usually refers to the process of extracting unique patterns and expertise [6], [15], [16]. The general purpose of text mining is to search for the latest information that has never been found before by processing and analyzing large amounts of data [17]. The Text Mining process starts with collecting documents. Then, text mining will select certain documents to be processed by checking the writing format and character set, which will then go through a text analysis stage that must be repeated several times until the expected information can be obtained [16].

Text Pre-Processing is carried out in the early stages of the mining process, a text mining method that aims to change data from unstructured to structured data for other mining processes. Text Pre-Processing is one of the crucial stages because it does not guarantee that the data to be processed is perfect, has no redundancy, and is consistent [6], [23]. Several stages of Text-Preprocessing will be used in this research, including case folding, tokenizing, stopwords, and stemming [6], [18], [19].

C. Cosine Similarity

The Cosine Similarity formula is used to measure the similarity value between two different pieces of data based on the results of the Text Mining process [20]. There are three similarity value measurement formulas other than Cosine Similarity to measure the similarity between two pieces of data, namely Jaccard Similarity, Overlap Coefficient, and Containment Measure. In this research, the Cosine Similarity formula is used to measure the similarity value because Cosine Similarity provides a higher threshold value when compared to the other three formulas [21]. The following is the Cosine Similarity formula used in the similarity calculation : [21]

$$similarity = \cos(\theta) = \frac{A.B}{\|A\|.\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

In calculating cosine similarity, two sentences are converted into terms/words. The words are converted to vector form, where each word defines a dimension in Euclidean space, and each word's frequency corresponds to the dimension's value. If the result of the similarity calculation is close to the value of one, then the sentences are said to be identical [18].

D. Cohen's Kappa

The statistical calculation method used to conduct reliability testing in this study is Cohen's Kappa. Cohen's Kappa is a statistical measure used to measure agreement between two or more observers in categorical data. Cohen's Kappa can also measure observer agreement in various types of categorical data, including data resulting from diagnostic tests, surveys, or data classification. The following is Cohen's Kappa calculation formula: [22]

$$K = \frac{p_0 - p_e}{1 - p_e} \quad (2)$$

After finding the calculation results, the results are then adjusted based on the kappa score interpretation table in Table 2:

TABLE I. INTERPRETATION OF THE KAPPA SCORE

Kappa Index	Agreement Proportion
< 0	less than chance-agreement
0,01 – 0,20	slight agreement
0,21 – 0,40	fair agreement
0,41 – 0,60	moderate agreement
0,61 – 0,80	substantial agreement
0,81 – 1	almost perfect

III. DATASETS

The data source used in this study consists of 27 documents labeled d1 - d27, which are taken from the elicitation results, Functional and Non-Functional Requirements. In the Requirement Elicitation dataset, initial identification is carried out by grouping the elicitation results taken from several questions and answers, which are separated based on the topic of discussion, which refers to the formation of Functional and Non-Functional Requirements (see Table 2). The other datasets used in this research are Functional and Non-Functional Requirements that have been labeled documents and also labeling references or their relationship with Requirement Elicitation that has been identified (see Table 2).

TABLE II. ARTEFACT DOCUMENT LABELING

ID	Document Labeling
RE-01	d1
RE-02	d2
RE-03	d3
RE-04	d4
RE-05	d5
RE-06	d6
RE-07	d7
RE-08	d8
RE-09	d9
FR-01	d10
FR-02	d11
FR-03	d12

FR-04	d13
FR-05	d14
FR-06	d15
FR-07	d16
FR-08	d17
FR-09	d18
FR-10	d19
FR-11	d20
FR-12	d21
FR-13	d22
NFR-01	d23
NFR-02	d24
NFR-03	d25
NFR-04	d26
NFR-05	d27

IV. METHODOLOGY

As shown in Figure 1, there are four main stages and activities presented in the research activity: topic analysis, similarity process, and reliability testing. The details of this design process are as follows:

A. Topic Analysis

At this stage, a topic analysis is carried out, which consists of three processes, namely:

1) Direct Requirement Elicitation activities by collecting information through discussions and interviews with user/stakeholders directly to equalize perceptions between users and developers.

2) Artifact modeling and document processing from interview transcripts and Functional and Non-Functional Requirement artifacts.

3) Text analysis process by applying the concept of Text Pre-Processing, which uses Python programming.

B. Similarity Process

At this stage, the Cosine Similarity formula is used to calculate the similarity value of the dataset in the Text Pre-Processing process. This value calculation is done with the help of the Python programming language.

C. Reliability Testing

This stage is to measure the reliability of the results of measuring the level of agreement between two observers from the questionnaire results for each Functional and Non-Functional Requirement artifact.

D. Final Result

At this stage, the value of the results of the similarity process and the kappa value from the effects of reliability testing on Requirement Elicitation and Requirement Specification (FR and NFR) are produced as a reference for validating FR and NFR.

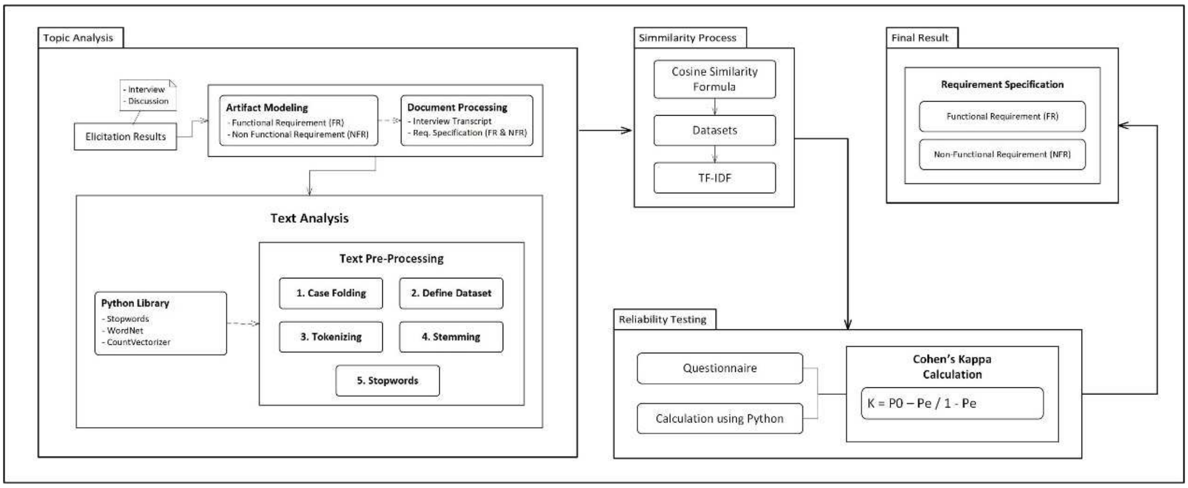

Fig. 1. Research Activity

V. RESULT AND DISCUSSION

The results of this study are taken based on the research methodology previously described, which refers to Figure 1 research activity: Topic Analysis, Similarity Process, Reliability Testing, and Final Results.

A. Topic Analysis

1) Requirement Elicitation: This process is carried out as an initial stage before the requirement specification is formed by means of a two-way direct interview between the application developer and the client. The results of this process are presented in Table 3. Interview topics are made into specification statements which are divided into two, namely:

a) Functional Requirement refers to the system specification and some functions that must be designed to meet user needs.

b) Non-Functional Requirement refers to the device specification and system architecture to support the quality and performance of the application to be created.

2) Artifact Modeling and Document Processing: From the previous stage, the artifact modeling stage is the process of creating and separating functional and non-functional requirements artifacts, which will then be used and labeled in the document processing stage. A document processing stage comprised 27 documents. Each document is labeled d1 - d27 (Table 4), which is taken from the elicitation, Functional Requirement, and Non-Functional Requirement results.

TABLE III. REQUIREMENT ELICITATION DOCUMENT

ID	Labelling Document	Requirement Elicitation Grouping	Identification
RE-01	d1	Discussion of menu request, cart, and transaction tracking features.	Hi, we want to follow up on the… etc
RE-02	d2	Discussion of payment methods in transactions.	Are there other features besides… etc
RE-03	d3	Discussion of data filter features, favorite products, and product reviews.	Anything else to add? We want to… etc
RE-04	d4	Discussion of features for communication.	For communication with customers … etc
RE-05	d5	Discussion of promo information and announcement features.	Do you have any additional information… etc
RE-06	d6	Discussion of authentication and account management features.	Do we need to use basic authentication… etc
RE-07	d7	Discussion of database specifications.	Who will use the app? The app will… etc
RE-08	d8	Discussion of server and operating system specifications.	Do we need high server specifications… etc
RE-09	d9	Discussion of Android smartphone specifications.	Can we adjust the specifications to … etc

Detail of requirement elicitation transcripts on "identification" column can be accessed on https://bit.ly/DatasetsRE.

TABLE IV. REQUIREMENT SPECIFICATION DOCUMENT

ID	Labelling Document	Requirement Specification Statement
FR-01	d10	Customers must register/login to their account on the Baker's Corner app.
FR-02	d11	Customers must be able to see promotional information banners when opening the app.
FR-03	d12	Customers must be able to see a list of recommended products by the store.
FR-04	d13	Customers must be able to filter data on the product list page.
FR-05	d14	Customers must be able to add products to their favorite product list.
FR-06	d15	Customers must be able to make menu requests on the menu request page.
FR-07	d16	Customers must be able to view order data on the cart page.
FR-08	d17	Customers must be able to make payments with linkaja or cash on the payment page.
FR-09	d18	Customers must be able to see transaction tracking on the transaction detail page.
FR-10	d19	Customers must be able to review products that have been ordered.
FR-11	d20	Customers must be able to communicate with the seller when opening the chat feature.
FR-12	d21	Customers must be able to see announcements made by the store owner when opening the app.
FR-13	d22	Customers must be able to change their profile information in the manage account feature.
NFR-01	d23	The system can only be run on smartphones with a minimum of 1 GB of RAM.
NFR-02	d24	The system can only be run on smartphones with a minimum android operating system version of 5.0 (Lollipop).
NFR-03	d25	The server system can be accessed on a minimum of Ubuntu Server 18.04 LTS with a minimum of 1 GB of RAM.
NFR-04	d26	The back-end server system is run on Nginx version 1.23.1 web server.
NFR-05	d27	The server system uses a SQL database that stores all product data and other supporting data.

3) Text Pre-Processing: After the document processing stage, here are the steps for text pre-processing with the help of the NLTK (Natural Language Toolkit) library provided by Python. Here are some of the processes performed:

a) Case folding, to convert all words in the sentence into lowercase letters in the dataset stored in the file "Elicitation_AfterCaseFolding.txt" (See Figure 2).

```
# Case Folding Req. Elicitation
file = open('Elicitation_Dataset_BeforeCase.txt', 'r')

lines = [line.lower() for line in file]
with open('Elicitation_AfterCaseFolding.txt', 'w') as out:
    out.writelines(sorted(lines))
print(lines)
```

Fig. 2. Case Folding

Case folding results can be accessed on https://bit.ly/CaseFoldingResults.

b) Tokenization, after converting all sentences into lowercase letters, word-by-word separation is carried out in the sentence using CountVectorizer (See Figure 3).

```
# Tokenization & Stop Words Elicitation
LemDocuments = CountVectorizer(tokenizer=word_tokenize, stop_words='english')
LemDocuments.fit_transform(ELICITATION_dataset)

print(LemDocuments.vocabulary_)
```

Fig. 3. Tokenization

Tokenization results can be accessed on https://bit.ly/TokenizationResults.

c) Stemming, this stage is carried out to convert words in the language into their basic form or base word and reduce variations in terms with the same form (See Figure 4).

```
# Stemming
def stemSentence(sentence):
    token_words = word_tokenize(sentence)
    token_words
    stem_sentence=[]
    for word in token_words:
        stem_sentence.append(porter.stem(word))
        stem_sentence.append(" ")
    return "".join(stem_sentence)

# Stemming ELICITATION
for i in range (len(ELICITATION_dataset)):
    ELICITATION_dataset[i] = stemSentence(ELICITATION_dataset[i])
    print (ELICITATION_dataset[i])
```

Fig. 4. Stemming

Stemming results can be accessed on https://bit.ly/StemmingResults.

d) Stop words used to remove words that are not used in the document. This process is carried out in conjunction with the CountVectorizer, which is carried out simultaneously in the tokenization process in Figure 3.

B. Similarity Process

After doing Text Pre-Processing, to calculate the similarity value, the TF-IDF (Term Frequency-Inverse Document Frequency) method is used to calculate the weight of words in the document and Cosine Similarity to calculate the level of similarity. Here are some stages of the similarity process:

1) Term Frequency (TF) process to calculate how important and often a word appears in the document (See Figure 5).

```
# Term Frequency (TF) ELICITATION
tf_matrixElicitation = LemDocuments.transform(ELICITATION_dataset).toarray()
print (tf_matrixElicitation)
```

Fig. 5. Term Frequency (TF)

Term Frequency results can be accessed on https://bit.ly/TermFrequencyResults.

2) Inverse Document Frequency (IDF) process, after getting the results of the Term Frequency process, proceed to the IDF process to measure how common a word is in all documents (See Figure 6).

```
# Inverse Document Frequency (IDF) ELICITATION
tfidfTranElicitation = TfidfTransformer(norm="l2")
tfidfTranElicitation.fit(tf_matrixElicitation)

print(tfidfTranElicitation.idf_)
```

Fig. 6. Inverse Document Frequency (IDF)

Inverse Document Frequency results can be accessed on https://bit.ly/IDFResults.

3) TF-IDF calculation, after the Term Frequency and Inverse Document Frequency process, the two matrices are combined for TF-IDF calculation (See Figure 7).

```
# TF-IDF Results Elicitation
tfidf_matrixElicitation = tfidfTranElicitation.transform(tf_matrixElicitation)
print (tfidf_matrixElicitation.toarray())
```

Fig. 7. TF-IDF

TF-IDF results can be accessed on https://bit.ly/TFIDFResults.

4) Calculation with Cosine Similarity, after obtaining the results of the TF-IDF process, then measuring the level of similarity between the two TF-IDF process vector results from the elicitation artifact and Requirement Specification (See Figure 8 and Figure 9).

	0	1	2	3	4	5	6	7	8
0	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
1	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
2	0,939723	0,999731	0,992446	0,44197	0,44197	0,44197	0,344939	0,075543	0,198953
3	0,625001	0,664912	0,660066	0,29395	0,29395	0,29395	0,229415	0,050243	0,132321
4	0,844467	0,971474	0,995655	0,2392	0,2392	0,2392	0,186686	0,040885	0,107676
5	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
6	0,264472	0,169453	0,120351	0,366277	0,366277	0,366277	0,285864	0,062605	0,16488
7	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
8	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
9	0,939723	0,999731	0,992446	0,44197	0,44197	0,44197	0,344939	0,075543	0,198953
10	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
11	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015
12	0,722056	0,462637	0,328578	1	1	1	0,780458	0,170924	0,45015

Fig. 8. Cosine Similarity FR & Elicitation

	0	1	2	3	4	5	6	7	8
0	0,196149	0,125677	0,08926	0,271654	0,271654	0,271654	0,307183	0,440776	0,736261
1	0,132186	0,084694	0,060152	0,183068	0,183068	0,183068	0,207012	0,180274	0,680872
2	0,123449	0,079096	0,056176	0,170968	0,170968	0,170968	0,408983	0,805423	0,530752
3	0,128938	0,082614	0,058675	0,178571	0,178571	0,178571	0,48973	0,673899	0,209499
4	0,400934	0,428452	0,426149	0,184427	0,184427	0,184427	0,292558	0,226814	0,08302

Fig. 9. Cosine Similarity NFR & Elicitation

C. *Reliability Testing*

Reliability testing uses Cohen's Kappa from the questionnaire results for each Functional and Non-Functional Requirement artifact. The following are the reliability testing results for FR and NFR, which can be seen in Table 5.

TABLE V. COHEN'S KAPPA SCORE FOR FR & NFR

FR/NFR	RE	Kappa Score
FR-01	RE-06	0.615384
FR-02	RE-05	0.523809
FR-03	RE-03	0.736842
FR-04	RE-03	0.400000
FR-05	RE-03	0.411764
FR-06	RE-01	0.375000
FR-07	RE-01	0.210526
FR-08	RE-02	0.199999
FR-09	RE-01	0.523809
FR-10	RE-03	0.347826
FR-11	RE-04	0.375000
FR-12	RE-05	0.615384
FR-13	RE-06	0.545454
NFR-01	RE-09	0.782608
NFR-02	RE-09	0.523809
NFR-03	RE-08	0.736842
NFR-04	RE-08	0.615384
NFR-05	RE-07	0.210526

Kappa scores are used as a metric to measure the level of agreement between two or more observers. The kappa score is calculated using a python library with measurements of two parameters between two observers of the questionnaire results for each artifact. Here is one of the program codes written to make the measurements:

```
# Kappa score for elicitation X FR
fr1_rater1 = [0, 1, 1, 1, 1, 1, 0, 1, 1, 1]
fr1_rater2 = [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]

fr1_kappa = cohen_kappa_score(fr1_rater1, fr1_rater2)
print("FR1 kappa score =>", fr1_kappa)

FR1 kappa score => 0.6153846154065088
```

Fig. 10. Kappa score calculation

D. *Final Results of FR and NFR Validation*

From several stages that have been carried out previously, here are the final results of the similarity value of the text analysis results on Requirement Elicitation, which is a reference for validating FR and NFR, which can be seen in Table 6. Based on the final results, there are two FR documents with ID FR-07 and FR-08 and one NFR document with ID NFR-05 that have been marked with orange color, which have low similarity values.

TABLE VI. FR & NFR VALIDATION RESULT AGAINST RE

FR/NFR	RE	Similarity Score	Kappa Score
FR-01	RE-06	1.000000	0.615384
FR-02	RE-05	1.000000	0.523809
FR-03	RE-03	0.992446	0.736842
FR-04	RE-03	0.660066	0.400000
FR-05	RE-03	0.995655	0.411764
FR-06	RE-01	0.722056	0.375000
FR-07	RE-01	0.264472	0.210526
FR-08	RE-02	0.462637	0.199999
FR-09	RE-01	0.722056	0.523809
FR-10	RE-03	0.992446	0.347826
FR-11	RE-04	1.000000	0.375000
FR-12	RE-05	1.000000	0.615384
FR-13	RE-06	1.000000	0.545454
NFR-01	RE-09	0.736261	0.782608
NFR-02	RE-09	0.680872	0.523809
NFR-03	RE-08	0.805423	0.736842
NFR-04	RE-08	0.673899	0.615384
NFR-05	RE-07	0.292558	0.210526

The reliability testing results using Cohen's Kappa show that FR-04, FR-06, FR-10, and FR-11 with the blue markers have values that fall into the "fair agreement" category (See Table 1). When viewed from the similarity value, some of these documents have a relatively high value but differ from the relatively low-reliability test value. This is due to the difference in interpretation between the two observers in assessing each document which results in relatively low values. But overall, the other documents have matched the reliability and similarity values.

TABLE VII. FACTORS CAUSING LOW SIMILARITY SCORE

FR/NFR	RE	Contributing factor
FR-07	RE-01	The RE results do not explicitly state that a cart page is required to view order data.
FR-08	RE-02	The RE results do not explicitly state that the payment method can use LinkAja or cash payment, but the results are still relevant even though it is not written explicitly.
NFR-05	RE-07	There are differences in terminology and terms, but they have the same meaning, such as "MySQL" with "SQL".

From the three documents with low similarity values, if a manual comparison is made of each FR / NFR document with the RE document, several factors cause the value to be low, such as differences in terminology and terms with the same meaning. In one of the documents being compared, it does not explicitly provide the same statement to make the two documents have a low similarity value. The following is an analysis of the causal factors of the three FR and NFR documents, which can be seen in Table 7.

After analyzing the factors that cause low similarity values in some of the documents above, improvements to FR and NFR documents can be seen in Table 8.

TABLE VIII. FR & NFR DOCUMENT IMPROVEMENT RESULTS

FR/NFR	Improvement Results
FR-07	Customers must be able to view order details before proceeding with the transaction process.
FR-08	Customers must be able to make payments with other online payment methods or cash on the payment page.
NFR-05	The server system use MySQL database that stores all product data and other supporting data.

VI. CONCLUSION AND FUTURE WORK

This research produces a total of 27 documents, each of which is labeled d1 - d27 taken from the results of RE as many as 9 documents (d1 - d9), FR as many as 13 papers (d10 - d22), and NFR as many as 5 documents (d23 - d27). From the reliability testing and similarity process results, it is shown that the proposed method can help validate the requirement specifications with the same results as the manual validation process. This method can provide practical guidance for software developers, researchers in the field of software requirements, and stakeholders who aim to improve the efficiency and accuracy of the requirement specification validation process. This method also aims to mitigate discrepancies in user and developer perceptions that can lead to requirement misalignment.

In future software specification process, it is necessary to improve FR and NFR by using the same terminology and terms in writing to ensure a higher similarity value. Then in the similarity calculation process, it should consider different terms with the same meaning so that it can still be calculated the same even with different terms. Future research may also implement this method to build an automatic validation tool that receives software specification and elicitation artefacts as inputs.

REFERENCES

[1] Arruda, D., Laigner, R., "Requirements Engineering Practices and Challenges in the Context of Big Data Software Development Projects: Early Insights from a Case Study," Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020 9377734, pp. 2012-2019, 2020.

[2] M. Asif, I. Ali, M. S. A. Malik, M. H. Chaudary, S. Tayyaba, and M. T. Mahmood, "Annotation of Software Requirements Specification (SRS), Extractions of Nonfunctional Requirements, and Measurement of Their Tradeoff," IEEE Access, vol. 7, pp. 36164–36176, 2019, doi: 10.1109/ACCESS.2019.2903133.

[3] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. M. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review," in 14th IEEE International Requirements Engineering Conference (RE'06), IEEE, Sep. 2006, pp. 179–188. doi: 10.1109/RE.2006.17.

[4] J. A. Pamungkas, Y. Priyadi, and M. J. Alibasa, "Measurement of Similarity Between Requirement Elicitation and Requirement Specification Using Text Pre-Processing in the Cinemaloka Application," in 2022 IEEE World AI IoT Congress (AIIoT), IEEE, Jun. 2022, doi: 10.1109/AIIoT54504.2022.9817193.

[5] Laliberte, C.D., Giachetti, R.E., Kolsch, M., "Evaluation of Natural Language Processing for Requirements Traceability," 2022 17th Annual System of Systems Engineering Conference, SOSE 2022 pp. 21-26, 2022.

[6] Naumcheva, M., "Object-Oriented Approach for Requirements Specification," CEUR Workshop Proceedings 3122, 2022.

[7] R. P. Octavially, Y. Priyadi, and S. Widowati, "Extraction of Activity Diagrams Based on Steps Performed in Use Case Description Using Text Mining (Case Study: SRS Myoffice Application)," IEEE, 3rd ICE3IS, 2022.

[8] C. M. Zapata J., B. M. Losada, and G. Gonzalez-Calderon, "An approach for using procedure manuals as a source for Requirements Elicitation," in 2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI), IEEE, Oct. 2012, pp. 1–8. doi: 10.1109/CLEI.2012.6426914.

[9] Coulentianos, M.J., Daly, S.R., Sienko, K.H., "Stakeholder perceptions of requirements elicitation interviews with and without prototypes in a cross-cultural design setting," Proceedings of the ASME Design Engineering Technical Conference 11B-2020, V11BT11A012, 2020.

[10] I. K. Raharjana, D. Siahaan, and C. Fatichah, "User Story Extraction from Online News for Software Requirements Elicitation: A Conceptual Model," in 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE, Jul. 2019, pp. 342–347. doi: 10.1109/JCSSE.2019.8864199.

[11] D. G. P. Putri and D. O. Siahaan, "Software feature extraction using infrequent feature extraction," in 2016 6th International Annual Engineering Seminar (InAES), IEEE, Aug. 2016, pp. 165–169. doi: 10.1109/INAES.2016.7821927.

[12] Y. Priyadi, A. M. Putra, and P. S. Lyanda, "The similarity of Elicitation Software Requirements Specification in Student Learning Applications of SMKN7 Baleendah Based on Use Case Diagrams Using Text Mining," in 2021 IEEE 5th International Conference on Information Technology, ICITISEE, IEEE, Nov. 2021, pp. 115–120. doi: 10.1109/ICITISEE53823.2021.9655844.

[13] M. Broy, "Rethinking Functional Requirements: A Novel Approach Categorizing System and Software Requirements," in Software Technology: 10 Years of Innovation in IEEE Computer, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018, pp. 155–187. doi: 10.1002/9781119174240.ch9.

[14] M. A. Kohl, K. Baum, M. Langer, D. Oster, T. Speith, and D. Bohlender, "Explainability as a Non-Functional Requirement," in 2019 IEEE 27th International Requirements Engineering Conference (RE), IEEE, Sep. 2019, pp. 363–368. doi: 10.1109/RE.2019.00046.

[15] A.-W. Tan, "Text Mining: The state of the art and the challenges," in Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases, 1999, pp. 65–70.

[16] Geng, B., "Legal Text Mining and Analysis Based on Artificial Intelligence," International Journal on Artificial Intelligence Tools 31(4),2240006, 2022.

[17] Wang, X., Tian, J., Li, F., "Text data mining of power based on natural language processing technology," Journal of Physics: Conference Series 2221(1),012050, 2022.

[18] Y. Priyadi, K. Kusumahadi, and P. S. Lyanda, "IdVar4CL Causal Loop Variable Identification Method for Systems Thinking Based on Text Mining Approach," International Journal of Fuzzy Logic and Intelligent Systems, 2022.

[19] A. S. Nayak and A. P. Kanive, "Survey on Pre-Processing Techniques for Text Mining," International Journal Of Engineering And Computer Science, Jun. 2016, doi: 10.18535/ijecs/v5i6.25.

[20] X. Wang, Z. Xu, X. Xia, and C. Mao, "Computing User Similarity by Combining SimRank++ and Cosine Similarities to Improve Collaborative Filtering," in 2017 14th Web Information Systems and Applications Conference (WISA), IEEE, Nov. 2017, pp. 205–210. doi: 10.1109/WISA.2017.22.

[21] P. P. Gokul, B. K. Akhil, and K. K. M. Shiva, "Sentence similarity detection in Malayalam language using cosine similarity," in 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, May 2017, pp. 221–225. doi: 10.1109/RTEICT.2017.8256590.

[22] A. S. Kolesnyk and N. F. Khairova, "Justification for the Use of Cohen's Kappa Statistic in Experimental Studies of NLP and Text Mining," Cybern Syst Anal, vol. 58, no. 2, pp. 280–288, Mar. 2022, doi: 10.1007/s10559-022-00460-3.

[23] Sakthi Vel, S., "Pre-Processing techniques of Text Mining using Computational Linguistics and Python Libraries," Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021 9395924, pp. 879-884, 2021.