informatics

MDPI

Article

A Novel Framework to Detect Irrelevant Software Requirements Based on MultiPhiLDA as the Topic Model

Daniel Siahaan *,† and Brian Rizqi Paradisiaca Darnoto

Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
* Correspondence: daniel@if.its.ac.id; Tel.:+62-31-5939214
† Current address: Gedung Informatika, Jl. Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111, Indonesia.

Abstract: Noise in requirements has been known to be a defect in software requirements specifications (SRS). Detecting defects at an early stage is crucial in the process of software development. Noise can be in the form of irrelevant requirements that are included within an SRS. A previous study had attempted to detect noise in SRS, in which noise was considered as an outlier. However, the resulting method only demonstrated a moderate reliability due to the overshadowing of unique actor words by unique action words in the topic–word distribution. In this study, we propose a framework to identify irrelevant requirements based on the MultiPhiLDA method. The proposed framework distinguishes the topic–word distribution of actor words and action words as two separate topic–word distributions with two multinomial probability functions. Weights are used to maintain a proportional contribution of actor and action words. We also explore the use of two outlier detection methods, namely percentile-based outlier detection (PBOD) and angle-based outlier detection (ABOD), to distinguish irrelevant requirements from relevant requirements. The experimental results show that the proposed framework was able to exhibit better performance than previous methods. Furthermore, the use of the combination of ABOD as the outlier detection method and topic coherence as the estimation approach to determine the optimal number of topics and iterations in the proposed framework outperformed the other combinations and obtained sensitivity, specificity, F1-score, and G-mean values of 0.59, 0.65, 0.62, and 0.62, respectively.

Keywords: angle-based outlier detection; percentile-based outlier detection; multiphilda; noise; irrelevant software requirements

Citation: Siahaan, D.; Darnoto, B.R.P. A Novel Framework to Detect Irrelevant Software Requirements Based on MultiPhiLDA as the Topic Model. *Informatics* **2022**, *9*, 87. https://doi.org/10.3390/informatics9040087

Academic Editors: Sanjay Misra, Robertas Damaševičius and Bharti Suri

Received: 27 August 2022
Accepted: 17 October 2022
Published: 27 October 2022

1. Introduction

The requirements specification process is one of the key stages in a software development project that determines its success. Researchers indicate that 40–60% of software development project failures originate in requirements specification [1,2]. Requirements dictate how the product is designed and implemented in the following stages [3]. Overlooking software requirements during the requirements specification process can cause future threats and failures in the operational phase [4]. Furthermore, the cost of detecting and correcting defects increases exponentially as the software progresses along the software development life cycle (SDLC) [5]. Therefore, it is essential and critical to carry out an effective requirements specification.

In an object-oriented approach, requirements engineers deliver an SRS, which is a document that serves as a guideline for the subsequent processes of software development [6]. Therefore, the SRS should have a set of quality attributes in order to maintain the quality of the end product. Nevertheless, requirements engineers often fail to comply with the quality attributes and produce ambiguity, contradictions, forward references, over-specifications, and noise in SRS [7,8]. Meyer [8] argues that the use of natural language to specify software requirements is the cause of these problems. This study focuses on noise, as it is a mistake

commonly made by requirements engineers. According to Meyer, noise refers to a condition where an SRS contains obscurity [9].

There are two types of noise in SRS, namely irrelevant requirements and non-requirement statements. Irrelevant requirements in SRS are requirements stated in the document that discusses topics beyond the scope of the software project. On the other hand, non-requirement statements in SRS are elements in the document that are of relevance to the domain of the software being developed but are not requirements. Manek and Siahaan [10] proposed a method to identify noise in SRS using spectral clustering. However, it was indicated in the study that the method was unable to effectively identify noise due to two reasons. Firstly, the method does not take into consideration the difference between irrelevant requirements and non-requirements statements. Secondly, the method disregards the different parts of requirements, namely actors and actions, during the identification process.

Natural Language Processing (NLP) has played an important role in human–computer interaction and artificial intelligence. It is often implemented in the domain of language structure analysis, technology-based machine translation, and language recognition [11]. Several unsupervised methods have been proposed to identify and model topics within textual documents. One of the most popular methods for topic modeling is Latent Dirichlet Allocation (LDA) [12,13]. The basic idea of the LDA method is that each document is represented as a random mixture of hidden topics, where each topic is characterized based on the distribution of words that lie within the topic itself. The LDA model generates a distribution of topics for the documents that can be immediately used as features for classification. The combination of topic and word frequency information may result in an enhanced set of features and in turn increase the performance of text classifiers [14]. Schröder [15] developed an LDA-based document classification method for accurate topic-based tagging of scientific articles. The proposed method is called CascadeLDA. It was indicated in the study that CascadeLDA required significantly fewer iterations compared to another extension of LDA, namely Labeled LDA (L-LDA), but it relies heavily on the label structure and general characteristics of the dataset. In the studies conducted by Suri and Roy [16] and Chen et al. [13], the LDA method was compared to Negative Matrix Factorization (NMF) for data detection with a large text flow. It was shown that the semantic results produced by LDA were more meaningful than the NMF method.

Irrelevant requirements can be viewed as outliers, since they discuss topics that are not within the scope of a system, in which the word components of irrelevant topics deviate from the word components of relevant topics. Several studies have proposed methods to detect outliers. Kriegel et al. [17] proposed the ABOD method for the detection of outliers. ABOD works on high-dimensional data. This method evaluates the degree of outliers based on the variance of the angle (VOA) formed by the target object and all other object pairs. The smaller the angular variance of the object, the more likely the object is an outlier. A recent study combined the EBOD method with several different methods and proposed an ensemble-based outlier detection (EBOD) method to detect outliers in noisy datasets [18]. In addition to ABOD, another method for outlier detection is the PBOD method. Sharma et al. [19] developed a PBOD method to reduce noise in large datasets and explored the use of machine learning methods, namely Gradient Boosting Regression (GBR) and Random Forest (RF), to predict short-term waiting times at the US–Mexico border. The results encourage incorporating more advanced prediction algorithms and methods into the dataset. Another research study went further and combined percentile and LDA to predict Days on Market, which is a measure of liquidity in the real estate industry [20].

This study is a continuation of our previous work [21]. The aim of this study is to identify noise in the form of irrelevant requirements within SRS documents. The main contribution of this study is proposing a new noise detection framework which relies on a modification of LDA, which is called MultiPhiLDA. The proposed framework distinguishes actors and actions as separate word distributions with two multinomial probability functions. Therefore, the process of clustering topics becomes more efficient due to the omission of irrelevant requirements within the SRS documents.

2. Related Works

Research in requirements engineering has received a tremendous amount of attention in the current epoch of predominant software industries. The concern in requirements engineering empirical investigation is growing significantly as a whole [22]. Almost one-fifth of the research focuses on analyzing the requirements, which include its quality. Detecting noise in requirements specification has also attracted attention from requirements engineering researchers [9,10,23,24]. Others focused on the impact of noise, such as defects [25–28], incorrect requirements [29], over-specification [30], and incompleteness [31,32].

Manek and Siahaan [10] proposed a method to detect noise in SRS using spectral clustering. The method was tested on 648 requirements that were manually extracted from 14 SRS documents. However, based on their experiment, the method showed moderate reliability. They suggested that this is due to the suboptimal pre-processing text process, in which they did not preserve the actors and actions in the requirement statements as a distinctive element of a requirement. Hence, actors or actions with noise can cause a requirement to become irrelevant.

Fahmi and Siahaan [9] compared the performance of several supervised methods in classifying non-requirement statements in SRS. They used the dataset that was previously used by Manek and Siahaan [10]. The classification methods that were analyzed were Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree, Random Forest (RF), and k-Nearest Neighbor (KNN). Their experiment showed that SVM produced a better classification model compared to the other methods to identify non-requirement statements in SRS.

Li et al. [33] developed a new model called the Common Semantics Topic Model (CSTM) to solve noise problems found in short texts and analyzed the performance of the model on three real-world datasets. They assumed that each short text is a mixture of the topic of the selected function and all common topics. According to Li, words that have high-frequency domains are a type of noise. CSTM defines a new type of topic to capture standard semantics and noise words.

Liu et al. [34] proposed Collaboratively Modeling and Embedding–Dirichlet Multinomial Mixtures (CME-DMM) to apply topic modeling on short texts that were collected from a social media platform. Liu identified major challenges in modeling latent topics in short texts, including the inadequacy of word co-occurrence instances, the need to capture local contextual dependencies in noisy short texts due to their limited length, and maintaining consistency between the latent topic models based on word co-occcurrences and their representation based on local contextual dependencies. The proposed method overcomes these challenges by considering both the global and contextual word co-occurrence relationship and also ensuring a high consistency between the topic and word embeddings in the latent semantic space.

Vo and Ock [35] classified short text documents using topic modeling. They used the LDA method before performing classification. They applied three machine learning methods, namely SVM, NB, and KNN for the classification of short text documents from three universal datasets and compared the performances of the three methods. The results using SVM are better in terms of accuracy compared to the other two methods.

Albalawi et al. [36] conducted a comparative analysis on the application of topic modeling methods on short text data. With the use of use of the LDA and Non-negative Matrix Factorization (NMF) methods, higher quality and coherence topics were obtained compared to the other methods. However, the LDA method was more flexible and provides more meaningful and logical extracted topics, especially with fewer topics. Both LDA and NMF showed similar performance, but LDA was more consistent. As a result, the LDA method outperformed other topic modeling methods with the highest number of features. The experimental results showed that LDA was able to obtain higher F-score values compared to the other methods. Furthermore, the LDA method was able to define the best and most precise meaning than the other topic modeling methods.

There have been previous works on identifying noise or non-requirement statements, but the classification performance is still poor. It fails to recognize that there are two types of noise, i.e., irrelevant requirements and non-requirement statements. This study addresses these two types of noise in requirements. Our study also proposed a new LDA-based topic modeling method called MultiphiLDA. It employs two separated distributions of words in a topic, namely actors and actions.

3. Noise Detection Framework

This section details the noise detection framework which relies on MultiPhiLDA to detect noise in SRS. Figure 1 shows that the input to the framework is a list of requirements which may be irrelevant or relevant requirements. In order to differentiate between irrelevant and relevant requirements, this framework has six main processes. This section elaborates these processes in more detail.

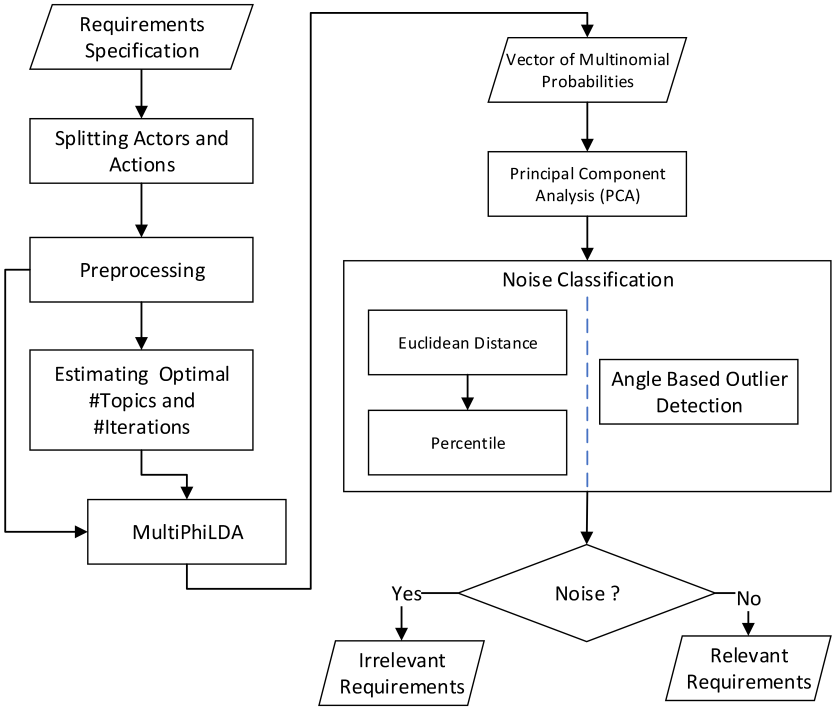

Figure 1. The noise detection framework.

3.1. Actor–Action Splitting

In the first step, the framework begins by identifying the actor and action parts within each requirement statement with the use of syntactic dependency parsing. In this study, we implemented a python code that loads the spacy library. Listing 1 shows how the spacy library was used for syntactic dependency parsing between words within a sentence. In the example, the sentence "All staff can view the details of any student." is parsed. A word with 'nsubj' tag would be classified as actor, while the rest would be classified as actions.

Figure 2 shows the result of the parsing process on the example sentence. The word 'staff' has a 'nsubj' dependency, which means that the word 'staff' is the subject who is doing the action and along with the determiner 'all' is identified as an actor part. The word 'view' has a 'ROOT' dependency, which means that the word 'view' and its direct object (the accusative object of 'view') is identified as an action part. After conducting syntactic dependeny parsing on each sentence, the action and actor parts are futher treated separately.

Listing 1. Extracting actors and actions from requirements statement.

```
#requirements:a textual dataset (contains requirement statements)
#[actors, actions]:lists of actors or actions extracted from~requirements

for requirement in data_skpl:
  postag = spaCy(requirement)
  for words in postag:
    if words.type == 'nsubj':
      actor.append(words)
    else:
      action.append(words)
```

```
All tag : det
staff tag : nsubj
can tag : aux
view tag : ROOT
the tag : det
details tag : dobj
of tag : prep
any tag : det
student tag : pobj
. tag : punct
```

Figure 2. Results of syntactic dependency parsing using the SpaCy Library.

3.2. Pre-Processing

In the second step, each action and actor part is then pre-processed. The pre-processing involves case folding, punctuation removal, tokenization, lemmatization, and stopwords removal. In the case folding process, all the words are normalized to lowercase. In the punctuation removal process, all numbers, symbols, and punctuations are removed. In the tokenization process, all the words are split individually into tokens. For example, the actor part of the example sentence "all staff" is split into two tokens 'all' and 'staff', while the action part of the example sentence "can view the details of any student" is split into six tokens 'can', 'view', 'the', 'details', 'of', 'any', and 'student'. After tokenization, lemmatization aims to reduce the words to their base form with the use of a vocabulary and morphological analysis. The last step in the pre-processing step is stopwords removal, in which non-substantial elements within a text, such as articles, conjunction, preposition, modals, and pronouns, are removed.

3.3. Optimal Number of Topics and Iterations Estimation

Before determining the optimal number of topics and iterations, unique words from every actor and action parts need to be extracted separately, and a set of actor words and a set of action words are obtained. Each set of unique words is stored as a separate dictionary of unique words. This process employs the gensim library to create the dictionaries. The dictionaries of unique words are later used to determine the optimal number of topics and the number of iterations in the application of MultiPhiLDA.

This study compares three estimation approaches to determine the optimal number of topics and interations, namely perplexity, topic coherence, and human judgment. Perplexity and topic coherence are commonly used to measure the performance of topic modeling methods [36,37]. Perplexity is considered to be a good approximation for a case where the test data are similar to training data. While perplexity only considers intrinsic measurements, topic coherence also consideres extrinsic measurements. These two measurements are supposed to make a distinction between semantically expoundable topics and factitious topics of statistical inference. In this study, we also compared the two estimation approaches against human judgement. Software engineers are involved in determining the optimal number of topics and iterations required to cluster the requirements based on its topics.

3.4. MultiPhiLDA

MultiPhiLDA is a modification of the LDA method. The LDA method has one phi, where phi is the distribution of words in a topic. Listing 2 shows the pseudocode of MultiPhiLDA. Previous research had proposed that requirements should be divided into two parts that separately represent actors and action, as these are two critical parts of a requirement. The number of unique actor words is significantly lower than the number of unique action words. If the topic–word distribution is based on unique words extracted from requirements and the topic–word distributions of action words and actor words are not defined separately, then the unique action words will overshadow the unique actor words in the topic–word distribution. Therefore, we need to create separate topic–word distributions for action words and actor words and subsequently amalgamate a new distribution from these two distributions by assigning weights to the two distributions to overcome the aforementioned problem within the new topic–word distribution. Figure 3 shows the plate notation of the MultiPhiLDA method. It can be seen in Figure 3 that phi is two, because MultiPhiLDA processes two topic–word distributions: one for actor words and the other for action words.

Listing 2. MultiPhiLDA pseudocode.

```
dataset = data[actors, actions]
for all actors
  randomnly assign a topic to the actor
end for
for all actions
  randomly assign a topic to the action
end for
for iteration
  for all actors in dataset
    for all actors
       theta calculation
       phi calculation
       phi calculation with weight = phi * actor_weight / Σ(phi * actor_weight)
       topic determination using phi and theta
    end for
  end for
  for all actions in dataset
    for all actions
      theta calculation
      phi calculation
      phi calculation with action weight = phi * action_weight / Σ(phi * action_weight)
      topic determination using phi and theta
    end for
  end for
end for
```

The proposed framework is expected to find irrelevant requirements in SRS. Requirements that have been processed and have topics assigned to them are further clustered with respect to topics. Clustering can be carried out using the K-means algorithm or other methods. The result of clustering is visualized to identify relevant and irrelevant requirements.

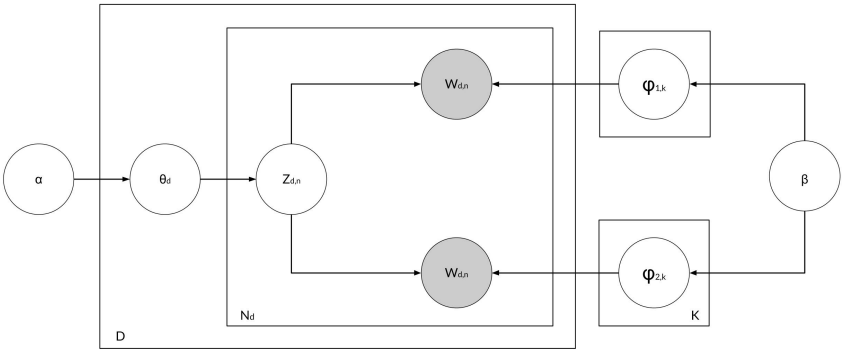

Figure 3. MultiPhiLDA plate notation.

Figure 4 illustrates the results of clustering by topic using MultiPhiLDA. The black dots represent relevant topics, while the white dots represent irrelevant topics. In an SRS, relevant requirements discuss relevant topics that are within the scope of the system, while irrelevant requirements discuss irrelevant topics that are outside the scope of the system. For example, a hospital information system may have relevant topics that are within the scope of the system such as patients, pharmacies, outpatients, poly, diagnosis, and disease. The visualization shown in Figure 4 can be used to indicate which topics are considered relevant or irrelevant.

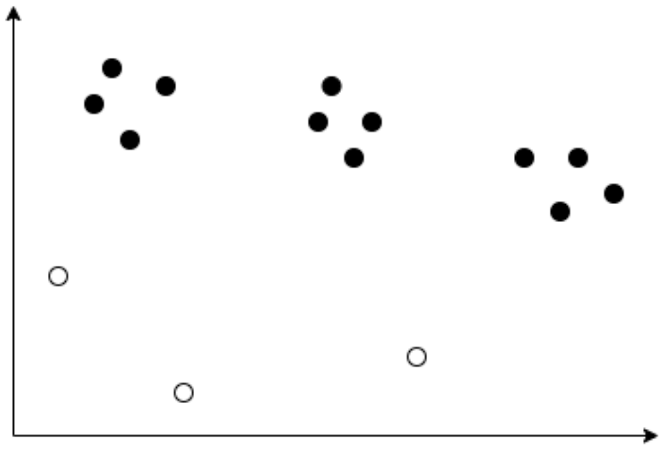

Figure 4. Visualization of MultiPhiLDA.

3.4.1. Separate Topic–Word Distribution for Actor Words and Action Words

After the identification of actor words and action words, two topic–word distributions are created, one for actor words and the other for action words. Firstly, the topics and the words are determined. Once the topics and words have been determined, we randomly assign topics to words. The probability of topics for a document is calculated using Equation (1), while the probability of words belonging to a topic is calculated using Equation (2) as follows:

$$\theta_{d,k} = \frac{n^{k}_{(d)} + \alpha}{n_{(d)} + K\alpha} \tag{1}$$

$$\phi_{k,v} = \frac{n^{v}_{(k)} + \beta}{n_{(k)} + V\beta} \tag{2}$$

The next step is to apply the weight for the actor words and action words. Here, the weight is used to determine the topic–word distribution for both sets of actor words and action words. The weights for action and actor words are predetermined, where the

sum of both weights is equal to 1. Once the phi value is obtained, weighting is carried out using Equation (3) as follows:

$$\phi = \frac{\phi_i \ weight}{\Sigma(\phi \ weight)} \tag{3}$$

Equation (4) is used to calculate the probability of a document belonging to a topic by taking into consideration the number of topics that the document may belong to. The likelihood of a word in the document belonging to a topic (w_n) is calculated by taking into consideration the number of topics that the word may belong to. For a specific w_n in document d, find the topic k whose probability has the maximum value and reassign the word to topic k.

$$p(z_i = k \mid z_i, w) = \phi_{k,v} \ \theta_{d,k} \tag{4}$$

Do as many iterations as previously estimated until two topic–word distributions of actor words and action words are obtained based on the predetermined number of topics.

3.4.2. Amalgamation of the Topic–Word Distributions

In this process, the two topic–word distributions of action words and actor words are merged into one topic–word distribution based on the topics that each word may belong to. Each topic consists of a few actor words and action words. The words that represent a topic are determined through their probabilistic values for the given topic. Amalgamation is carried out by sorting the probabilistic values of action and actor words for a given topic, and only a few words with the highest probabilistic values are selected to represent the topic. After amalgamating the two distributions into a single distribution, the next process is to identify the irrelevant requirements in the SRS documents.

3.5. Identification of Irrelevant Requirements

This process discriminates the requirements statements that are irrelevant. In this study, we compared the performance of two approaches, namely PBOD and ABOD. Percentiles are values below a certain percentage of observations [38]. In this study, we calculated the percentile value using Euclidean distance. Euclidean distance calculates the average distance between each member of a cluster to their centroid. If the value is greater than the percentile value, then the requirement statements that belongs to the cluster are considered irrelevant requirements.

ABOD is a method to detect outliers. ABOD assigns an Angle-Based Outlier Factor (ABOF) to each point in the database and returns a list of points that had been sorted based on the ABOF of the points [17]. Figure 5 illustrates the ranking of each points within a dataset. The top-ranking point (number 1) is the farthest outlier. The subsequent ranks are determined by the next farthest cluster boundary point. Since distance is only taken into account as a weight for the variance of angles, ABOD can succinctly detect outliers even in high-dimensional data.

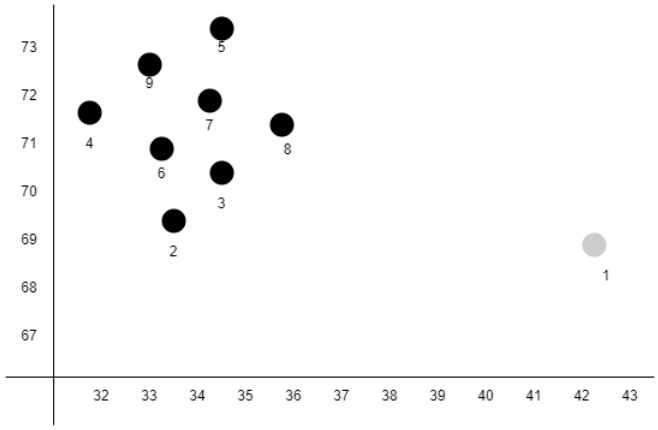

Figure 5. Point ranking in the sample dataset.

4. Experimental Setup

This section explains how the experiment was carried out in this study. The first section describes the dataset used in the experiment. The second second describes the three different evaluation scenarios used to evaluate the performance of the proposed framework. These three scenarios are performed as a means of providing empirical data to indicate the performance of the proposed framework.

4.1. Dataset

In this study, we used the same dataset that was used in previous research conducted by Manek and Siahaan [10]. The dataset is publicly available [39]. The dataset contains fourteen SRS from various software development projects. However, the original dataset contains typos and grammatical errors. Therefore, we proofread each document beforehand. Table 1 shows the statistical information of each SRS document within the dataset. The number of requirement statements (#Reqs) varies highly between documents (between 6 and 106). The average length of each requirement statement (the average number of words, µword) also varies. The average number of verbs (µverb) in a requirement statement is between one and two words. The SRS documents of the different projects possess different numbers of actors. It can also be seen that the dataset is highly imbalanced, where the number of irrelevant requirements (#Irr) is significantly smaller (9.26%) than the number of relevant requirements.

Table 1. Recapitulation of the Requirements in Each SRS.

ID	Project Name	#Reqs	#μ-word	#μ-verb	#Actor	#Irr
DA-1	Submit Job	13	6	1	1	1
DA-2	System Administrator	17	12	1	4	5
DA-3	Archive Administrator	39	12	2	4	2
DA-4	SPG Application	6	6	1	1	3
DA-5	System Administrator Staff	33	40	1	8	3
DA-6	Software Development	65	17	2	33	5
DA-7	Display System	106	13	2	17	3
DA-8	Internet Access	64	18	2	22	7
DA-9	Meeting Initiator	33	24	1	7	4
DA-10	Online System	17	15	1	7	2
DA-11	Library System	86	17	2	32	9
DA-12	IMSETY System	70	12	2	13	3
DA-13	Manage Student Information	24	22	1	9	3
DA-14	PHP Project	75	28	1	50	10

4.2. Evaluation Scenarios

In this study, we conducted three evaluation scenarios to evaluate the performance of the proposed framework. Scenario 1 aims to evaluate the topics produced by the proposed framework and estimate the optimal number of topics and iterations. Scenario 2 aims to measure the performance of the proposed framework using four performance metrics, namely specificity, sensitivity, F1-score, and G-mean. Sensitivity is the ratio of true positive predictions to the total number of actual positive data. Specificity is the ratio of true negative predictions to the total number of actual negative data. G-mean combines specificity and sensitivity and balances both concerns. F1-score combines precision and recall values produced by the proposed framework by taking their harmonic mean. Scenario 3 aims to measure the reliability of the proposed framework by using two reliability measurements, namely average variance extracted (AVE) and composite reliability (CR) to evaluate the consistency of the proposed framework in identifying irrelevant requirements over the fourteen SRS documents. In all the evaluation scenarios, all possible combinations of estimation approaches (perplexity, topic coherence, and human judgment) and outlier

detection methods (ABOD and PBOD) within the proposed framework were individually evaluated. Furthermore, in scenario 2, a comparison of performance was carried out between the proposed framework and the method proposed by Manek and Siahaan [10].

5. Result

In this section, we evaluated the performance of the proposed framework based on the three evaluation scenarios.

5.1. Evaluation Scenario 1: Estimate the Optimal Number of Topics and Iterations

This scenario aims to evaluate the topics produced by the proposed framework with the use of different estimation approaches (perplexity, topic coherence, and software engineer) and estimate the optimal number of topics and iterations. In this evaluation, the minimum number of topics for each SRS is two, while the maximum number of topics is predetermined by the software engineer. Furthermore, in this evaluation, the minimum number of iterations is one, while the maximum number of iterations is 500. The evaluation was carried out on two models for each estimation approach: one that uses ABOD and another that uses PBOD as the outlier detection method. The values obtained from the two models are added and divided by two. After obtaining one principal value, the value is visualized to estimate the optimal number of topics and iterations. Figure 6 shows the perplexity scores and the coherence scores obtained from the two models, which are added and divided by two.

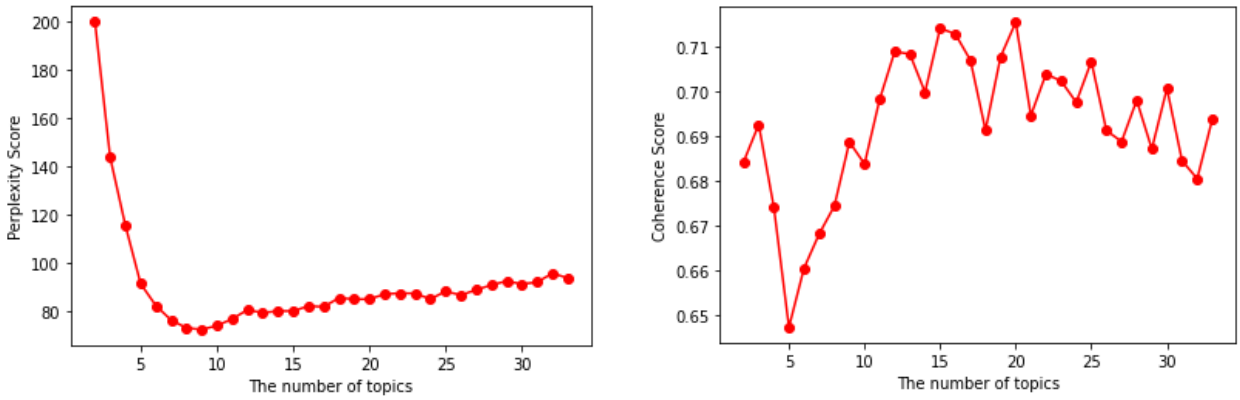

Figure 6. Visualization of topic coherence and perplexity values.

With the use of topic coherence, the most satisfactory model is indicated by the highest coherence score. On the other hand, with the use of perplexity, the most satisfactory model is indicated by the lowest perplexity score. The results of the optimal number of topics and iterations can be seen in Table 2.

After obtaining the optimal number of topics, the next step is to determine the optimal number of iterations with the use of the optimal number of topics that had been estimated by the three estimation approaches. The SRS document used to determine the optimal number of iterations must have a graph of perplexity values whose ups and downs are not too extreme and possess a gentle trend value. If the SRS document does not meet these requirements, the SRS document cannot be used. Figure 7 is an example of an SRS document (DA-07) with a graph of perplexity values and trend values that meet these conditions. The number of requirements within the DA-07 SRS document is the largest among all the SRS documents.

Table 2. The Optimal Number of Topics and Number of Iterations for Each SRS Document.

Noise Classifier	Perplexity		Topic Coherence		Software Engineer		Status
	#Topic	# Iteration	#Topic	#Iteration	#Topic	#Iteration	
DA-01	5	224	4	273	4	273	Not Accepted
DA-02	15	62	13	493	9	424	Not Accepted
DA-03	10	425	19	490	9	59	Not Accepted
DA-04	5	22	4	323	6	62	Not Accepted
DA-05	18	466	12	286	19	86	Not Accepted
DA-06	23	198	12	473	27	201	Not Accepted
DA-07	9	122	21	203	35	187	Accepted
DA-08	14	74	20	123	26	400	Not Accepted
DA-09	13	96	18	294	18	294	Not Accepted
DA-10	7	400	6	107	11	239	Not Accepted
DA-11	19	433	30	390	36	132	Not Accepted
DA-12	12	330	9	375	48	430	Not Accepted
DA-13	17	472	10	409	20	500	Not Accepted
DA-14	23	430	39	89	53	320	Not Accepted

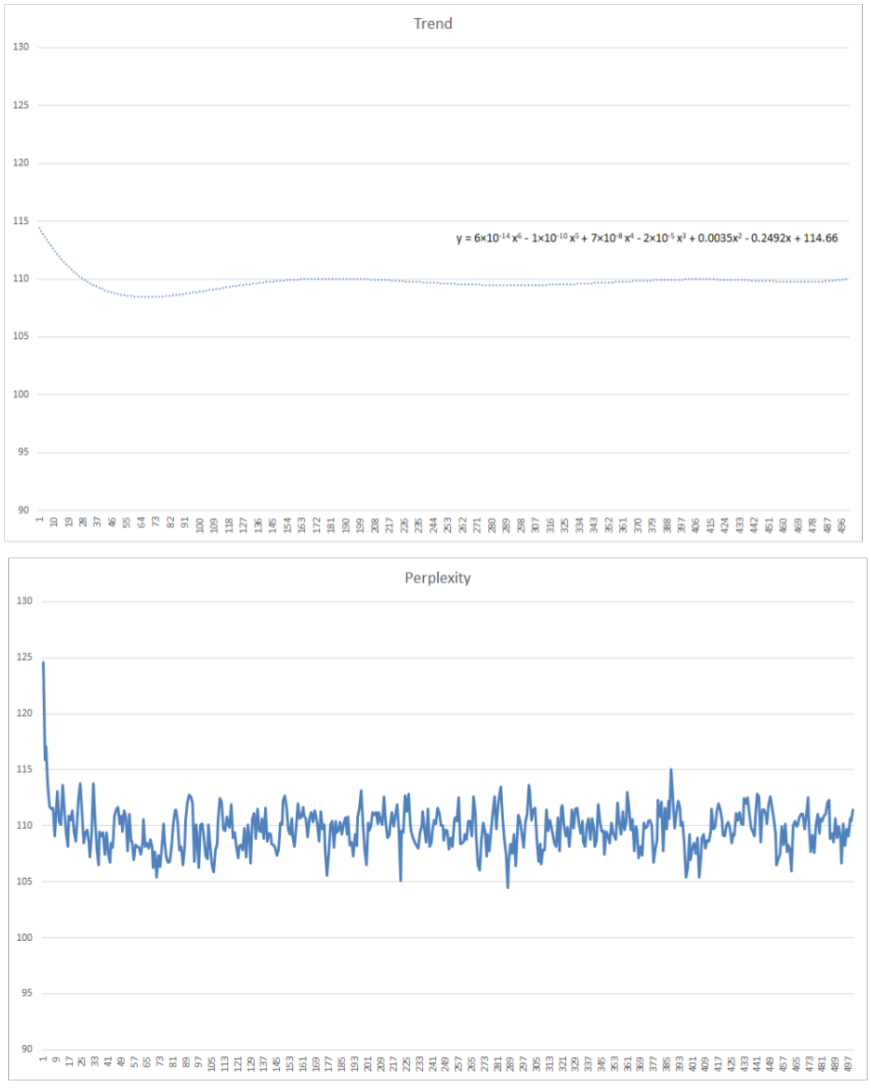

Figure 7. Accepted perplexity value and trend value.

5.2. Evaluation Scenario 2: Sensitivity, Specificity, F1-Score, and G-Mean

In this scenario, we compared the alternative combination of three estimation approaches and two outlier detection methods which used in the proposed framework in terms of sensitivity, specificity, F1-score, and G-mean. Six models were evaluated, which were constructed by using different combinations of estimation approaches (perplexity, topic coherence, and software engineer) and outlier detection methods (PBOD and ABOD). Since the dataset is highly imbalanced, in which around 80–90% of the requirements in each document are relevant requirements, we used the performance metrics, namely sensitivity, specificity, F1-score, and G-mean, to evaluate the six models. Table 3 shows the evaluation results of the six models. Each score is calculated over all fourteen SRS documents based on a weighted average that takes into consideration the number of requirements in each SRS document. This is because according to Tang et al. [40], the number of documents, in this case requirements, plays an important role in the performance of a model to identify topics. Furthermore, we also compared the performance of the six models against the method proposed by Manek and Siahaan [10].

Table 3. Comparison of Performance between the Six Models and Previous Study.

Noise Classifier		Sensitivity	Specificity	F1-Score	G-Mean
PBOD	Perplexity	0.28	0.81	0.54	0.47
	Topic Coherence	0.27	0.80	0.53	0.46
	Software Engineer	0.23	0.82	0.52	0.43
ABOD	Perplexity	0.43	0.70	0.56	0.55
	Topic Coherence	0.59	0.65	0.62	0.62
	Software Engineer	0.60	0.56	0.58	0.58
Manek and Siahaan [10]		0.10	0.65	0.38	0.25

Based on Table 3, the use of human judgment to estimate the optimal number of topics and iterations resulted in the highest sensitivity and specificity values, namely 0.60 and 0.82, respectively. However, the overall best model is the one that uses topic coherence as the estimation approach and ABOD as the outlier detection method. The ability of the proposed framework to correctly detect irrelevant requirements (true positive rate) and relevant requirements (false positive rate) is significantly higher than that of the method proposed by Manek and Siahaan [10].

5.3. Evaluation Scenario 3: Reliability of the Proposed Framework

In this scenario, we evaluated the reliability of each combination (estimation approach and outlier detection method). In this evaluation, we applied two reliability measurements, namely average variance extracted (AVE) and composite reliability (CR). Both metrics were used to measure the consistency of each model to identify irrelevant requirements over the fourteen SRS documents (see Table 4).

Table 4. Sensitivity and Specificity Values on the DA-07 SRS Document.

Noise Classifier		Sensitivity	Specificity	AVE	CR
PBOD	Perplexity	0.28	0.81	0.03	0.75
	Topic Coherence	0.27	0.80	0.03	0.77
	Software Engineer	0.23	0.82	0.02	0.74
ABOD	Perplexity	0.43	0.70	0.04	0.85
	Topic Coherence	0.59	0.65	0.05	0.93
	Software Engineer	0.60	0.56	0.05	0.93

Based on the experimental results, the six models exhibited considerably consistent performance over all the SRS documents. Regardless of the outlier detection method, the use of human judgment as the estimation approach produces reliable results in terms of both AVE and CR. With the use of ABOD, the model always produces CR values higher than 0.8, which is the threshold value commonly used to measure reliability [41]. This indicates that the noise classifier that uses ABOD as the outlier detection method is very reliable. In addition, the best combination to be used in the proposed framework is ABOD as the outlier detection method and either topic coherence or human judgment as the estimation approach.

6. Discussion

Although the results of evaluation scenario 2 indicate that the combination of ABOD and topic coherence exhibits better performance in terms of F1-score and G-mean than the other combinations, the use of human judgment as the estimation approach to determine the optimal number of topics and iterations combined with ABOD also exhibits considerably good performance. The results of evaluation scenario 3 support this statement. We can see in Figure 8 that the combination of software engineer and ABOD exhibits a close pattern of G-mean scores compared to the combination of topic coherence and ABOD.

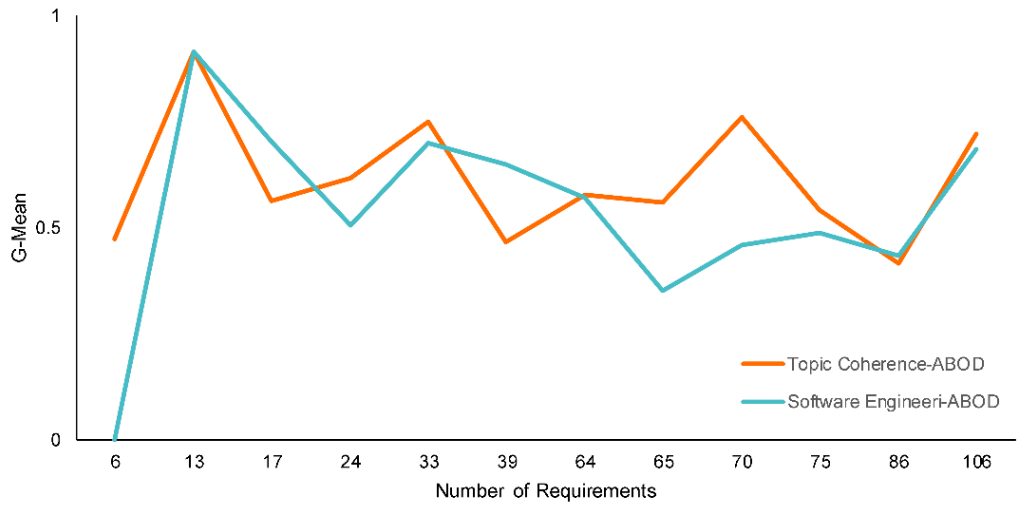

Figure 8. Performance of the two best models with respect to number of requirements in each SRS document.

Tang et al. [40] suggested that the number of topics may affect the efficacy of LDA in mapping the topics against documents. Figure 9 shows that MultiPhiLDA also supports this claim. Figure 9 represents the two-dimensional plotting between the topic ratio (the number of predicted optimal topics divided by the number of requirements within an SRS document) and the G-mean score. The trendline indicates that with more topics, the G-mean scores obtained by the proposed framework are lower. This phenomenon can be explained as follows. An SRS of a project that covers a wider problem domain may have more topics compared to one that covers a narrower problem domain. For instance, an enterprise resource planning (ERP) system covers a wider domain than a supply chain management (SCM) system. Therefore, the ERP system should have more topics than the SCM system. Another example is between an employee attendance (EA) system and a human resource development (HRD) system, where an EA system covers a narrower domain than an HRD system. Therefore, an EA system should have less topics than an HRD system. An HRD system may have subsystems that cover a broad range of topics related to attendance (clock in, clock out, schedule, paid leave, and overtime), career (role, grade, skills, retire), training (short-term, long-term, hardskill, softskill, and registration), payroll (salary, bonus, gross pay, work hours, exempt employee, pension, insurance, tax,

and allowance), recruitment (part-time, full-time, employee, contract, and application), and performance (target, realization, indicators, evaluation, planning, and monitoring). A topic within a subdomain should be relevant to the other topics in that particular subdomain. For instance, the clock-in topic is relevant to the other topics within the attendance system, such as clock-out, paid leave, and overtime. On the other hand, topics of different subdomains, for example clock-in and insurance, are usually irrelevant to each other. However, there may be an instance where a topic may be relevant to other topics in other subdomains. For example, the salary topic, which is part of the payroll system, may be relevant to topics in the attendance system such as paid leave and overtime. This is because both subdomains are part of a bigger domain.

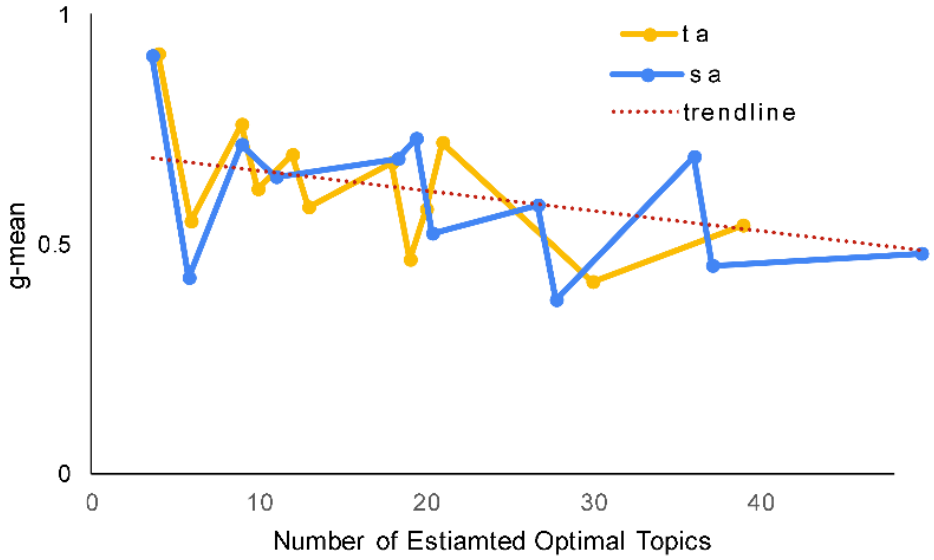

Figure 9. Performance of the two best models with respect to the number of requirements in an SRS document.

In order to further analyze the diagnostic ability of our proposed framework in detecting irrelevant requirements, we also conducted receiver operating characteristic (ROC) analysis on the six models of different combinations. Figure 10 shows the ROC curves of each model that has a different combination of estimation approaches and outlier detection methods. The curve is a plot between the true positive rate (TPR) and false positive rate (FPR) obtained by the model on each SRS document. A good classifier is indicated by a larger number of dots on the upper side of the diagonal line, while a bad classifier is indicated by a larger number of dots on the lower side of the diagonal line. From the six ROC curves, we can visually analyze that the combination of ABOD and topic coherence produced a higher number of dots on the upper side of the diagonal line compared to the other combinations. This indicates that the combination of ABOD and topic coherence results in a a better noise classifier to detect irrelevant requirements within SRS documents. The three dots that lie on the lower side of the diagonal line are associated to the DA-03, DA-04, and DA-11 SRS documents, in which the ratio between the estimated number of topics and the number of requirements is relatively higher than that of the other SRS documents.

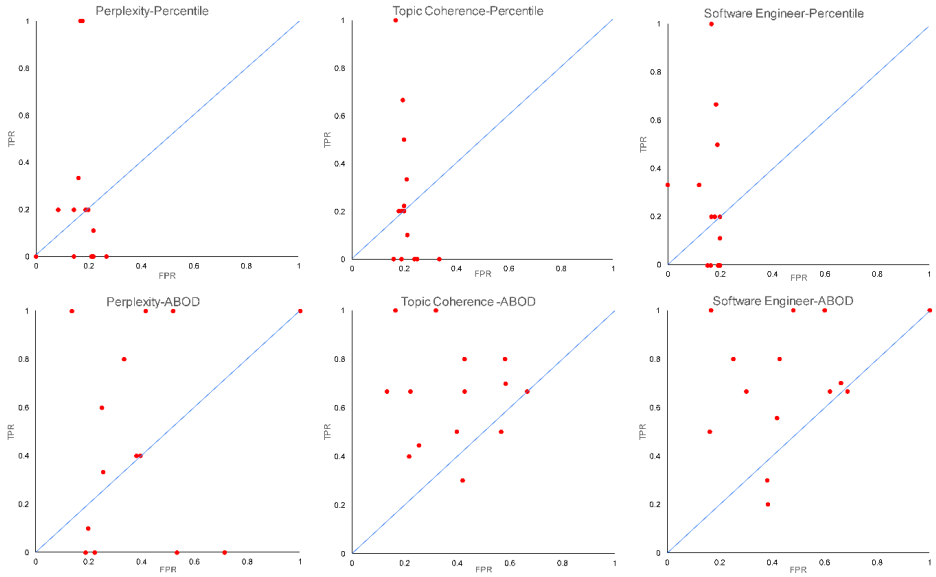

Figure 10. The ROC curve of the six models with different combinations of estimation approach and outlier detection method.

7. Conclusions

This study proposes a novel framework based on the MultiPhiLDA method to detect irrelevant requirements within SRS documents. The MultiPhiLDA method was developed based on the LDA algorithm. In the proposed framework, an estimation is carried out to estimate the optimal number of topics and iterations to be used in MultiPhiLDA as the topic modeling method, and also outlier detection is conducted to identify irrelevant requirements. In this study, we explored the use of three estimation approaches, namely perplexity, topic coherence, and human judgment. Furthermore, we explored the use of two outlier detection methods, namely PBOD and ABOD. We evaluated the performance of the proposed framework with different combinations of estimation approaches and outlier detection methods.

The experimental results indicate that the proposed framework exhibits better performance in detecting irrelevant requirements compared to previous methods. The proposed framework exhibits a better performance with the use of topic coherence as the estimation approach and ABOD as the outlier detection method, in which it obtained a sensitivity, specificity, F1-score, and G-mean value of 0.59, 0.65, 0.62, and 0.62, respectively. Furthermore, in terms of reliability, the use of the combination of ABOD and topic coherence for the proposed framework obtained the highest AVE and CR score of 0.05 and 0.93, respectively.

For future research, we will carry out optimization of the weight given to the topic–word distribution of action words and actor words. In this study, the weights of the topic–word distributions were manually predefined. We gave a greater weight to action words because we assumed there is a larger number of unique action words compared to unique actor words in a requirement. However, it is possible that the number of unique actor words is equal or even higher than the number of action words in a requirement. Therefore, it is necessary to carry out optimization of this particular model parameter to enhance the performance of the model to classify relevant and irrelevant requirements. Furthermore, we will also use projects that have a larger number of requirements in the experiment in order to obtain stronger empirical evidence in regard to the claim that more topics tend to decrease the performance of the proposed framework.

Another direction of future research would be investigating the computational complexity of the proposed framework. Current solution inherits a cubic-time complexity problem. An alternative solution would be the use of a heuristic approximation variant of the method, dimension reduction, or topic modeling algorithm that performs better on a small number of topics, small number of words in documents, and limited documents.

Author Contributions: Conceptualization, D.S. and B.R.P.D.; methodology, D.S.; software, B.R.P.D.; validation, D.S. and B.R.P.D.; formal analysis, D.S.; investigation, D.S. and B.R.P.D.; resources, B.R.P.D.; data curation, B.R.P.D.; writing—original draft preparation, D.S.; writing—review and editing, D.S. and B.R.P.D.; visualization, D.S. and B.R.P.D.; supervision, D.S.; project administration, D.S.; funding acquisition, D.S.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education, Culture, Research and Technology Republic of Indonesia grant number 084/E5/PG.02.00.PT/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset is available publicly by figshare at the following link (all accessed on 8 March 2022) https://doi.org/10.6084/m9.figshare.19380545.v1.

Acknowledgments: The authors thank the Institut Teknologi Sepuluh Nopember for the support of this research by providing a doctoral scholarship for Brian Rizqi Paradisiaca Darnoto.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABOD	Angle-Based Outlier Detection
CME-DMM	Collaboratively Modeling and Embedding–Dirichlet Multinomial Mixtures
CSTM	Common Semantics Topic Model
DT	Decision Tree
EBOD	Ensemble-Based Outlier Detection
HSLDA	Hierarchically Supervised LDA
KNN	k-Nearest Neighbor
LDA	Latent Dirichlet Allocation
NB	Naïve Bayes
NLP	Natural Language Processing
PBOD	Percentile-Based Outlier Detection (PBOD)
RF	Random Forest (RF)
SDLC	Software Development Life Cycle
SRS	Software Requirements Specifications
SVM	Support Vector Machine
VOA	Variance of the Angle

References

1. Iqbal, J.; Ahmad, R.B.; Khan, M.; Fazal-E-Amin; Alyahya, S.; Nasir, M.H.N.; Akhunzada, A.; Shoaib, M. Requirements engineering issues causing software development outsourcing failure. *PLoS ONE* **2020**, *15*, e0229785. [CrossRef] [PubMed]
2. Mandal, A.; Pal, S. Identifying the Reasons for Software Project Failure and Some of their Proposed Remedial through BRIDGE Process Models. *Int. J. Comput. Sci. Eng.* **2015**, *3*, 118–126.
3. del Sagrado, J.; del Águila, I.M. Stability prediction of the software requirements specification. *Softw. Qual. J.* **2018**, *26*, 585–605. [CrossRef]
4. Dalal, S.; Chhillar, R. Case studies of most common and severe types of software system failure. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2012**, *2*, 341–347.
5. Wiegers, K.E.; Beatty, J. *Software Requirements*, 3rd ed.; Microsoft Press: Washington, DC, USA, 2013; pp. 19–22.
6. Yanuarifiani, A.P.; Chua, F.F.; Chan, G.Y. An ontology framework for generating requirements specification. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2020**, *10*, 1137–1142. [CrossRef]
7. Zheng, L. Effective information elicit for software quality specification based on ontology. In Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, IEEE Computer Society, Beijing, China, 23–25 September 2015; pp. 677–680. [CrossRef]
8. Meyer, B. On Formalism in Specification. *IEEE Softw.* **1985**, *2*, 6–26. [CrossRef]
9. Fahmi, A.A.; Siahaan, D.O. Algorithms Comparison For Non- Requirements Classification Using The Semantic Feature of Software. *IPTEK J. Technol. Sci.* **2020**, *31*, 1–9. [CrossRef]

10. Manek, P.G.; Siahaan, D. Noise Detection In Software Requirements Specification Document Using Spectral Clustering. *JUTI J. Ilm. Teknol. Inf.* **2019**, *17*, 30–37. [CrossRef]
11. Wang, D.; Su, J.; Yu, H. Feature extraction and analysis of natural language processing for deep learning english language. *IEEE Access* **2020**, *8*, 46335–46345. [CrossRef]
12. Jelodar, H.; Wang, Y.; Yuan, C.; Feng, X.; Jiang, X.; Li, Y.; Zhao, L. Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey. *Multimed. Tools Appl.* **2019**, *78*, 15169–15211. Available online: http://xxx.lanl.gov/abs/1711.04305 (accessed on 13 June 2022). [CrossRef]
13. Chen, Y.; Zhang, H.; Liu, R.; Ye, Z.; Lin, J. Experimental explorations on short text topic mining between LDA and NMF based Schemes. *Knowl.-Based Syst.* **2019**, *163*, 1–13. [CrossRef]
14. Chen, Y.H.; Li, S.F. Using latent Dirichlet allocation to improve text classification performance of support vector machine. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation, CEC, Vancouver, BC, Canada, 24–29 July 2016; Institute of Electrical and Electronics Engineers Inc.: Hoes Lane Piscataway, NJ, USA, 2016; pp. 1280–1286. [CrossRef]
15. Schröder, K. Hierarchical Multiclass Topic Modelling with Prior Knowledge. Master's Thesis, Humboldt—Universität zu Berlin, Berlin, Germany, 2018.
16. Suri, P.; Roy, N.R. Comparison between LDA & NMF for event-detection from large text stream data. In Proceedings of the 3rd IEEE International Conference on "Computational Intelligence and Communication Technology", Ghaziabad, India, 9–10 February 2017; Institute of Electrical and Electronics Engineers Inc.: Hoes Lane Piscataway, NJ, USA, 2017; pp. 1–5. [CrossRef]
17. Kriegel, H.P.; Schubert, M.; Zimek, A. Angle-based outlier detection in high-dimensional data. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 444–452. [CrossRef]
18. Ouyang, B.; Song, Y.; Li, Y.; Sant, G.; Bauchy, M. EBOD: An ensemble-based outlier detection algorithm for noisy datasets. *Knowl.-Based Syst.* **2021**, *231*, 107400. [CrossRef]
19. Sharma, S.; Kang, D.H.; Montes de Oca, J.R.; Mudgal, A. Machine learning methods for commercial vehicle wait time prediction at a border crossing. *Res. Transp. Econ.* **2021**, *89*, 101034. [CrossRef]
20. Kamara, A.F.; Chen, E.; Liu, Q.; Pan, Z. A hybrid neural network for predicting Days on Market a measure of liquidity in real estate industry. *Knowl.-Based Syst.* **2020**, *208*, 106417. [CrossRef]
21. Darnoto, B.R.P.; Siahaan, D. MultiPhiLDA for Detection Irrelevant Software Requirement Specification. In Proceedings of the 2021 International Conference on Computer Science, Information Technology and Electrical Engineering, Banyuwangi, Indonesia, 27–28 October 2021; pp. 92–97. [CrossRef]
22. Ambreen, T.; Ikram, N.; Usman, M.; Niazi, M. Empirical research in requirements engineering: Trends and opportunities. *Requir. Eng.* **2018**, *23*, 63–95. [CrossRef]
23. Romano, S.; Scanniello, G.; Fucci, D.; Juristo, N.; Turhan, B. Poster: The effect of noise on requirements comprehension. In Proceedings of the International Conference on Software Engineering, Madrid, Spain, 23–29 September 2018; IEEE Computer Society: Los Alamitos, CA, USA, 2018; pp. 308–309. [CrossRef]
24. Jørgensen, M.; Grimstad, S. The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *IEEE Trans. Softw. Eng.* **2011**, *37*, 695–707. [CrossRef]
25. Ferrari, A.; Gori, G.; Rosadini, B.; Trotta, I.; Bacherini, S.; Fantechi, A.; Gnesi, S. Detecting requirements defects with NLP patterns: An industrial experience in the railway domain. *Empir. Softw. Eng.* **2018**, *23*, 3684–3733. [CrossRef]
26. Kamalrudin, M.; Ow, L.L.; Sidek, S. Requirements defects techniques in requirements analysis: A review. *J. Telecommun. Electron. Comput. Eng.* **2018**, *10*, 47–51.
27. Rosmadi, N.A.; Ahmad, S.; Abdullah, N. The relevance of software requirement defect management to improve requirements and product quality: A systematic literature review. *Adv. Intell. Syst. Comput.* **2015**, *355*, 95–106. [CrossRef]
28. Alshazly, A.A.; Elfatatry, A.M.; Abougabal, M.S. Detecting defects in software requirements specification. *Alex. Eng. J.* **2014**, *53*, 513–527. [CrossRef]
29. Chari, K.; Agrawal, M. Impact of incorrect and new requirements on waterfall software project outcomes. *Empir. Softw. Eng.* **2018**, *23*, 165–185. [CrossRef]
30. Martínez, A.B.B.; Arias, J.J.P.; Vilas, A.F. Merging requirements views with incompleteness and inconsistency. In Proceedings of the Australian Software Engineering Conference, ASWEC, Brisbane, Australia, 29 March–1 April 2005; pp. 58–67. [CrossRef]
31. Elgammal, A.; Papazoglou, M.; Krämer, B.; Constantinescu, C. Design for Customization: A New Paradigm for Product-Service System Development. *Procedia CIRP* **2017**, *64*, 345–350. [CrossRef]
32. Özlem Albayrak.; Kurtoğlu, H.; Biçakçi, M. Incomplete software requirements and assumptions made by software engineers. In Proceedings of the Asia-Pacific Software Engineering Conference, APSEC, Penang, Malaysia, 1–3 December 2009. [CrossRef]
33. Li, X.; Wang, Y.; Zhang, A.; Li, C.; Chi, J.; Ouyang, J. Filtering out the noise in short text topic modeling. *Inf. Sci.* **2018**, *456*, 83–96. [CrossRef]
34. Liu, Z.; Qin, T.; Chen, K.J.; Li, Y. Collaboratively Modeling and Embedding of Latent Topics for Short Texts. *IEEE Access* **2020**, *8*, 99141–99153. [CrossRef]
35. Vo, D.T.; Ock, C.Y. Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Syst. Appl.* **2015**, *42*, 1684–1698. [CrossRef]

36. Albalawi, R.; Yeap, T.H.; Benyoucef, M. Using Topic Modeling Methods for Short-Text Data: A Comparative Analysis. *Front. Artif. Intell.* **2020**, *3*, 42. [CrossRef]
37. Lubis, F.F.; Rosmansyah, Y.; Supangkat, S.H. Topic discovery of online course reviews using LDA with leveraging reviews helpfulness. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 426–438. [CrossRef]
38. Bornmann, L.; Leydesdorff, L.; Mutz, R. The use of percentiles and percentile rank classes in the analysis of bibliometric data: Opportunities and limits. *J. Inf.* **2013**, *7*, 158–165. Available online: http://xxx.lanl.gov/abs/1211.0381 (accessed on 13 June 2022). [CrossRef]
39. Siahaan, D. Irrelevant Requirements, 2022. Available online: http://dx.doi.org/10.6084/m9.figshare.19380545.v1 (accessed on 13 June 2022). [CrossRef]
40. Tang, J.; Meng, Z.; Nguyen, X.L.; Mel, Q.; Zhang, M. Understanding the limiting factors of topic modeling via posterior contraction analysis. In Proceedings of the 31st International Conference on Machine Learning, International Machine Learning Society (IMLS) 2014, Beijing, China, 21–26 June 2014; pp. 337–345.
41. Netemeyer, R.G.; Bearden, W.O.; Sharma, S. *Scaling Procedures: Issues and Applications*, 1st ed.; SAGE: London, UK, 2003; pp. 153–159.