**NLP FOR CONTRACT REVIEW**

Natural language processing (NLP) is widely used for contract review, as it can help identify relevant information and extract key data from contracts quickly and accurately. Here are some examples of how NLP can be used for contract review:

1. **Contract summarization**: NLP algorithms can be used to generate a summary of a contract, highlighting key clauses and terms, and providing a high-level overview of the document.
2. **Contract comparison**: NLP algorithms can be used to compare two contracts, highlighting differences between them and identifying areas where changes have been made.
3. **Entity extraction**: NLP algorithms can be used to extract important entities such as parties, obligations, and deadlines from contracts, making it easier to identify and track these elements.
4. **Clause identification**: NLP algorithms can be used to identify and extract specific clauses within a contract, such as termination clauses or indemnification clauses.
5. **Contract classification**: NLP algorithms can be used to classify contracts into different categories based on their contents, such as employment contracts, lease agreements, or purchase orders.
6. **Risk assessment**: NLP algorithms can be used to analyze contracts for potential risks, such as non-compliance with regulations, unclear obligations, or ambiguous terms.

Overall, NLP can be a powerful tool for contract review, helping to streamline the process and make it more efficient and accurate.

**PAPER**

[**https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa\_token=\_L9OCyA\_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg\_33wktDVLMwcJYz0SAfg**](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)

**Abstract**

Contract documents are a critical legal component of a construction project that specify all wishes and expectations of the owner toward the design, construction, and handover of a project. Precise comprehension of the contract documents is critical to ensure that all important contractual requirements of the project scope are captured and managed. A contract package typically includes both requirements and other unimportant texts such as instructions and supporting statements; thus, practitioners are required to read and identify texts indicating the requirements. The conventional manual practice of scope comprehension requires much time and effort and may include human errors. Little attention has been paid toward automated identification of requirement texts. This study introduces an effective way to identify contractual requirements by developing an automated framework using natural language processing (NLP) and machine learning techniques. Four different machine learning algorithms, namely Naïve Bayes, support vector machines, logistic regression, and feedforward neural network were used to develop the classification models. The models classified the contractual text into requirement and nonrequirement text. Experiments showed that the support vector machine model outperforms the other models in terms of accuracy, precision, recall, and F1-score.

**Introduction**

Requirement identification is a critical step of the project scope comprehension task. The contract package of a construction project typically includes a large amount of text that includes both unimportant texts (i.e., instructions and supporting statements) and critical texts (i.e., contractual requirements). This requires professionals to carefully read the whole package and identify texts indicating the requirements ([Le et al. 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). The keyword “shall” would be helpful to some extent in identifying the requirements. However, contract writers also use words other than “shall” that do not clearly convey the intended meaning. These ambiguous requirements may include undesirable provisions enforced by the client. In addition, many requirements including those referencing standards or regulatory codes are implicitly mentioned in the contract. Such implicit requirements are likely to be missed during the project scope comprehension stage ([Huovila and Seren 1998](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). The process of extracting requirements is very tedious and error-prone if relying on a human being because project contracts are often voluminous and ambiguous ([Le et al. 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)).

To fill this gap, this paper introduces a new domain specific classification model for the identification and extraction of client requirements from construction contracts. The study uses natural language processing (NLP) and machine learning to develop an automated framework for the classification of contractual text into two categories, namely requirements and nonrequirements. Various popular supervised machine learning algorithms with different settings were tested, and their performance was compared. Moreover, the effectiveness of the automated requirement detection model was further evaluated by conducting an experimental study that compares the performance between human and machine for the requirement identification task.

**Requirements Identification from Construction Contracts**

The precise comprehension of the client requirements presented in the contracts is crucial to avoid disputes among the contracting parties ([Rameezdeen and Rodrigo 2013](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Many studies have reported that the failure to identify the complete client requirements in the contracts is the most prevalent cause of disputes in construction projects ([Kilian 2003](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg); [Zhang and Liu 2012](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Contradictory and incomplete requirements in the contracts are several other key factors causing disputes ([Luxford 1998](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg); [Cheung and Pang 2013](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Such requirements, if not identified early, result in expensive delays and disputes. Because the construction contracts have become complex over time ([Bunni 2003](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)), the professionals who regularly perform the contract review task also often overlook a few client requirements that may affect the understanding of the project scope and obligations ([Walsh 2017](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). It generally happens when the bidders are required to perform the detailed contract review along with the preparation of technical and financial estimates in the short bidding period ([Lee et al. 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). In addition, the misinterpretation of the contractual requirements may result in a significant loss to the construction firms that put the financial conditions of the firm at risk ([Walsh 2017](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). For instance, a minor mistake in interpretation of the contractual requirements caused a Canadian firm to lose CAD$2.13 million ([Walsh 2017](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). This implies the need and importance of knowing the accurate client requirements before engaging in a contract.

**Natural Language Processing**

Text classification is a subdomain of NLP that aims to categorize the text into one or more predefined classes ([Manning et al. 1999](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). It is mainly performed using the following two approaches: ruled-based methods, and machine learning–based methods. Rule-based methods are more robust due to human involvement in the development of rules but require much effort to construct all rules manually ([Manning et al. 2009](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). In contrast, machine learning–based methods learn from the experience by training ([Manning et al. 2009](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg); [Kim and Chi 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Text classification can either be a single-label text classification or a multilabel text classification ([Tsoumakas and Katakis 2007](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). In single-label text classification, only one label can be assigned to each text sample. With multilabel text classification, it is possible for each text sample to be assigned with two or more labels ([Sebastiani 2002](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). A single label text classification can further be classified into two categories: (1) binary text classification, if only two classes are included in dataset; or (2) multiclass text classification, if more than two classes are involved ([Hotho et al. 2005](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)).

**Methodology**

The study presented in this paper has attempted to develop an automated framework for requirements identification from the contract documents. The developed system can classify the contractual text into two predefined categories, namely requirements and nonrequirements. Requirements refer to the important text describing the obligations and compulsions whereas nonrequirements refer to the instructions and supporting statements such as definitions and headings. First, a dataset was prepared to train and evaluate the models. Following this, the text classification was performed using two different approaches including rule-based classification and machine learning-based classification. In the rule-based method, several handcrafted rules were generated to optimize the performance of rule-based model. In machine learning-based approach, four supervised machine learning algorithms were implemented to analyze and compare the performance of requirement identification models. Finally, the highest performance model was assessed on an unseen contract document in an experimental study to compare the performance of manual and automated requirement identification process. The following sections provide the details of the dataset and the two approaches adopted in this study.

**Dataset Preparation**

For the purpose of the study, a dataset of labeled data was developed. A collection of 1,787 statements were extracted from seven different contract documents. These statements were manually labeled as being in either the requirement or nonrequirement category. The final dataset used in the study included 1,388 requirement statements and 399 nonrequirement statements. While labeled data is easily accessible for many other applications, such as classification of news articles ([Nigam et al. 1999](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)), data collection for the text classification for construction applications is comparatively difficult. However, the results of the text classification for the construction applications using small-size data are still reliable due to the availability of well-standardized data in construction ([Salama and El-Gohary 2016](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)).

During the data cleaning stage, undesired contents (table of contents, headers, footers, etc.) in the contract documents were first removed. Each statement in the contract documents was then annotated with requirement or nonrequirement labels according to the content of the statement. This annotated document was then structured to a single comma-separated values (CSV) file where the statements were manually labeled with their predefined classes.

**Rule-Based Classification**

Rule-based classification includes a set of “IF-THEN” rules in which the input part after IF shows the condition while the output part after THEN shows the conclusion ([Mladenov et al. 2014](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg))

IF  *condition*  THEN  *conclusion*

In order to develop the rules, a list of keywords for the requirement category was first prepared. Such keywords discriminating the requirement statements from nonrequirement statements were identified by carefully analyzing the most frequent words in the two categories. The words that appeared with high frequency in the requirement category but were not present in the nonrequirement category were considered to develop the rules for the requirement detection. Each keyword formed one rule that predicted the label of test statement based on the presence and absence of that keyword. The rules were added one by one to the model, and the classification performance was analyzed with each increment. The rules that improved the accuracy were retained while those reducing the accuracy were excluded from the final model. A total of 15 rules were found to improve the accuracy and were included in the final rule-based model. These final rules included the following keywords: “shall,” “integrated management system (IMS),” “require,” “follow,” “submittals,” “Submittals,” “system,” “final,” “minimum,” “required,” “accordance,” “submit,” “conformance,” “submittal,” and “requires.”

The statements satisfying any of the rules are labeled as requirements while the statements without any keywords in rules are considered nonrequirements. The predicted labels are then compared with the actual labels to determine the accuracy of the rule-based model.

**Machine Learning–Based Classification**

The proposed methodology uses several preprocessing techniques and feature representation methods to examine the improvement in classification results. Initial experiments were conducted using four different machine learning algorithms to determine their best performance.

**Data Preprocessing**

Prior to feature selection, the following data preprocessing techniques were implemented in this study:

|  |  |
| --- | --- |
| • | **Lowercasing and Punctuation Removal**: Lowercasing converted the complete text into a lowercase form to ensure that the similar meaning words such as “Construction” and “construction” are treated as one term. The punctuation marks were also removed as they did not contribute toward the text classification. |
| • | **Stop Words Removal**: Stop words are the frequent words in any statement that have less meaning and discriminative power, e.g., “the,” “and,” “is,” “for,” “of” ([Manning et al. 1999](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). In this study, all the English stop words were removed from the corpus to reduce the noise in the model. |
| • | **Unique Words Removal**: Unique words or the very low-frequency words that occurred only once in the whole corpus were also removed in preprocessing as they did not carry any discriminative power. |
| • | **Tokenization**: Tokenization split the complete extensive text into the single units or tokens that can be a single word, a punctuation mark, a number, or a white space. |
| • | **Lemmatization and Stemming**: Lemmatization and stemming converted all words in the corpus to their dictionary form by removing inflectional endings; for instance, “works,” “worked,” and “working” were reduced to the root word “work” ([Porter 1980](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Lemmatization can also match the synonym words such as trucks and automobiles by analyzing the vocabulary and morphology of the word. Stemming and lemmatization enhance the accuracy and computational efficiency of the model as now the model uses only one index to signify *different* grammatical forms of a word ([Balakrishnan and Ethel 2014](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |
| • | **Parts-of-speech (POS) Tagging**: The tokens were assigned the POS tags which represented the functional and lexical category of the terms or tokens, e.g., the terms building and stairs were tagged as (NN) singular noun, while the terms appointed and brick-built were tagged as (JJ) adjective, and the terms had and has were tagged as (VB) verb ([Cutting et al. 1992](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |

**Sentence Representation**

Because machine learning algorithms cannot directly process the words of a statement, it is typically converted into a numerical vector of the most relevant features ([Liu et al. 2017](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). A feature hereby refers to an attribute or word of a textual statement that can present its semantics. The following are two approaches to constructing representation vectors implemented in this study.

**Bag-of-Words Model**

Bag-of-words model (BOW) is the simplest as well as the most commonly used method to represent the text as numerical value features. The method considers the whole corpus as a bag of words ignoring the grammar and order of words. Using the BOW model, each statement in the corpus is represented as a numeric vector where each element in the vector corresponds to a word in the corpus. The values in the vector can either be zero, indicating the absence of a word in the statement, or a real number, indicating the frequency of the word in the statement ([Boulis and Ostendorf 2005](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). In this study, the frequency of a word is determined by using term frequency–invert document frequency (TF-IDF) which is the product of two parameters: (1) term frequency (TF), which indicates the frequency of a word in a specific statement; and (2) invert document frequency (IDF), which is a parameter that weighs down the high-frequency domain specific words and scales up the rare terms to determine the actual important terms in the corpus ([Sebastiani 2002](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). This study used Eq. ([1](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)) to compute the TF-IDF score of each word

(1)

TF−IDF=*ntN*×(1+log*KKt*)

where *nt* = number of occurrences of a term *t* in a statement; *N* = total number of words in the statement; *K* = total number of statements; and *Kt* = number of statements that include the term *t*.

In this study, the BOW model was further extended to bag of *n*

-grams to obtain the partial information related to the position of words in a statement ([Joulin et al. 2017](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). *N*-grams represent a sequence of co-occurring words as a single token for the text classification. *N*

-grams of different lengths of sequences such as unigrams (one word as one token), bigrams (two co-occurring words as one token), and trigrams (three co-occurring words as one token) were used for the training process. For instance, in the sentence “road widening is needed,” the unigrams are “road,” “widening,” “is,” and “needed” while bigrams are “road widening,” “widening is,” and “is needed,” and trigrams are “road widening is” and “widening is needed.”

**Word2vec Model**

In addition to the BOW method, Word2vec was also implemented for features representation. The extra semantic information provided by Word2vec representation may help in text classification ([Lilleberg et al. 2015](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Word2vec employs the artificial neural networks to generate a multidimensional vector that represents the semantics of every unique word in the whole corpus ([Kim and Chi 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Unlike the BOW representation where each statement is represented by a vector; each word is represented by a vector in Word2vec representation. Thus, a statement is represented as an array of representation vectors of all words in it. This set of word vectors represents the semantics of a statement that is used as the input data of the machine learning process. The word representation vectors are typically mapped on a high dimensional vector space of which the number of dimensions is determined by the number of nodes in the hidden layer ([Mikolov et al. 2013](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Word2vec considers the assumption that words appearing in the same context in the corpus have similar meanings ([Kim and Chi 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Therefore, the vectors of the similar sense words are brought closer in the vector space. This study employed the continuous bag-of-words (CBOW) algorithm for the Word2vec representation. CBOW algorithm predicts the current word considering the vectors of the surrounding words as input.

**Machine Learning Algorithms Implementation**

Before the implementation of the algorithms, the dataset of 1,787 statements were randomly split into training and testing datasets. The 1,250 statements (70%) were allocated in the training dataset, while 537 statements (30%) were assigned to the testing dataset. After dataset distribution, four popular supervised machine learning algorithms were implemented to develop the requirement identification models. Because each algorithm produces promising results based on the dataset and the domain, no particular algorithm can be considered as superior to others in all cases.

|  |  |
| --- | --- |
| • | **Naïve Bayes (NB)**: NB is a simple and user-friendly probabilistic algorithm ([Rennie et al. 2003](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). It employs the Bayesian theorem with a strong assumption that every feature is independent of other features, given a class label. Therefore, the discrete nature of features increases the computational efficiency of the algorithm ([Qiu et al. 2010](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |
| • | **Support Vector Machines (SVM)**: SVM works on the principle of establishing a hyperplane that splits the points of positive training data and negative training data with the maximum gap ([Joachims 1998](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)) and is known for its high performance for text classification problems ([Abu Sheikha and Inkpen 2010](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). The availability of kernel functions in SVM further allows it to classify the nonlinear data ([Chung et al. 2007](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |
| • | **Logistic Regression (LR)**: LR is a statistical machine learning technique that develops a correlation between discrete categorical dependent variables and a set of independent variables ([Bilal et al. 2016](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). It is a complex form of linear regression. This probabilistic approach is capable of predicting the probability of any data for the predefined categories ([Hosmer and Lemeshow 2000](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |
| • | **Feedforward Neural Network (FNN)**: FNN includes three layers, namely input layer, hidden or perceptron layers, and the output layer ([Nii et al. 2007](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). The input layer accepts the text statements, the hidden layer assigns weights to those text statements, and the output layer predicts the category of the statements. Neural networks are trained by the standard backpropagation learning rule ([Manevitz and Yousef 2007](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). Backpropagation is the iterative process, which continuously trained the classifier by minimizing the error in every iteration using an optimizer. These errors are determined by using a loss function, i.e., cross entropy function ([Grzegorczyk 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). |

**Classification Model Evaluation**

The classification models were evaluated using different performance metrics including accuracy, precision, recall, and F1-score [as shown in Eqs. ([2](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg))–([5](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)), respectively]. These performance metrics are determined based on the following figures: true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*), where *TPs* are the number of statements labeled correctly as “requirement,” *FPs* are the number of statements labeled incorrectly as “requirement,” *TNs* are the number of statements labeled correctly as “nonrequirement,” and *FNs* are the number of statements labeled incorrectly as “nonrequirement.” In plain language, precision shows the percentage of the true requirements among the total number of statements labeled as requirements by the model while recall indicates the percentage of true requirements in the testing dataset that are successfully detected by the model. Because precision and recall are typically two dual performance metrics ([Buckland and Gey 1994](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)), F1-score is commonly used to as a combined metrics of both precision and recall to assess the effectiveness of the model ([Abu Sheikha and Inkpen 2010](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg))

(2)

Accuracy=*TP*+*TNTP*+*TN*+*FP*+*FN*

(3)

Precision=*TPTP*+*FP*

(4)

Recall=*TPTP*+*FN*

(5)

F1*-*score=2×Precision×RecallPrecision+Recall

**Classification Performance**

This section presents the performance of rule-based and machine learning–based approaches considering different parameter settings. The performance of the machine-learning–based classification models was examined using different preprocessing techniques. In addition, the effect of *n*-grams, the removal of common frequent words, and the integration of POS information on the machine learning based classification were examined.

**Performance of Rule-Based Approach**

The rule-based classification was performed on the dataset by gradually adding rule by rule to the model. The change in the accuracy was observed when more rules were added. The improvement in accuracy with the addition of each rule is shown in Fig. [4](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg). The first rule developed using the keyword “shall” contributed the most toward the rule-based classification. The implementation of first rule resulted in an accuracy of 84.7%, which reflects the importance of this keyword in requirement detection. The next two rules improved the performance by 4.1%; however, minor improvement was observed with the addition of every next rule. The last 12 rules collectively increased the performance by just 1.8%. The maximum accuracy achieved by considering all 15 rules was 90.6%.

**Effect of Preprocessing Techniques on Classification Performance**

First, the performance of the four different machine learning algorithms were evaluated using different preprocessing techniques. The purpose of this experimentation was to select the best performance techniques for the subsequent analyses. The results revealed that all the preprocessing techniques discussed in the earlier section improved the classification accuracy except the stemming. Because the lemmatization and stemming performed the same task, a comparison was developed to examine the performance of the two techniques. Based on the results, it can be noted that lemmatization performed better than stemming for all algorithms. The maximum difference in performance was observed with the NB algorithm where the lemmatization showed 1.68% higher accuracy than stemming. For the other algorithms, the difference in classification performance using the two techniques was noted as less than 1%. Due to the better performance in text classification, lemmatization technique was selected for all subsequent comparative evaluations in this paper. Furthermore, SVM was revealed as the best algorithm for unigram text classification using lemmatization with an accuracy of 98.15%. NB, FNN, LR, and Word2vec+LR also achieved the comparable accuracy results with 95.53%, 97.60%, 97.95%, and 96.27% accuracy values, respectively.

Table 1. Classification results using lemmatization and stemming with unigram features

| **Machine learning algorithms** | **Accuracy using lemmatization (%)** | **Accuracy using stemming (%)** |
| --- | --- | --- |
| NB | 95.53 | 93.85 |
| SVM | 98.51 | 98.32 |
| FNN | 97.60 | 97.20 |
| LR | 97.95 | 97.76 |
| Word2vec + LR | 96.27 | 95.53 |

Other performance measures including precision, recall, and F1-score for the both classes using lemmatization and unigram features are also reported in Table [2](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg). As discussed earlier, precision and recall are often two dueling performance metrics. Because the objective of this study was to minimize the missing of important contractual requirements detected, the recall value for the requirement class is most critical among all the performance measures. A recall value of 100% was desired, which reflects the successful identification of all the requirement statements in the testing corpus by the model. As shown in Table [2](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg), SVM exhibited relatively robust performance in terms of precision, recall and F1-score for the both classes. SVM reached a recall value of 95% for requirements, which is very close to the ideal value of 100% recall. In addition, the LR algorithm resulted in a recall value of 94% for requirements, which is higher than the remaining three algorithms. NB, FNN and Word2vec+LR yielded a recall of 84%, 93%, and 88%, respectively, for requirements. In terms of precision, SVM, LR, and Word2vec+LR showed the top three results with 98%, 97%, and 95% precision values, respectively.

Table 2. Performance measures of different unigram classification models using lemmatization

| **Machine learning algorithms** | **Class label** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
| --- | --- | --- | --- | --- |
| NB | Requirements | 94 | 84 | 89 |
| Nonrequirements | 96 | 99 | 97 |  |
| SVM | Requirements | 98 | 95 | 96 |
| Nonrequirements | 99 | 100 | 99 |  |
| FNN | Requirements | 94 | 93 | 94 |
| Nonrequirements | 98 | 99 | 98 |  |
| LR | Requirements | 97 | 94 | 96 |
| Nonrequirements | 98 | 99 | 99 |  |
| Word2vec + LR | Requirements | 95 | 88 | 91 |
| Nonrequirements | 97 | 99 | 98 |  |

**Effect of *n*-grams on Classification Performance**

*N*-gram models use a sequence of *n* co-occurring words rather than a single word as unit features in vector representation. Experiments were performed using different sizes of *n*-grams and the variation in classification accuracy was observed. Unigrams showed the highest accuracy and it consistently performed the best for all classifiers. Bigrams and trigrams did not improve the results. For *n*-gram of sizes higher than 3, the situation was similar. Results for sequences up to a maximum size of 3 are presented in Fig. [5](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg). As shown, SVM yielded the best performance for unigrams, bigrams, and trigrams with accuracy values of 98.51%, 97.39%, and 93.48%, respectively. For all classifiers, the results indicate that increasing the size or length of *n*-gram may not be useful and can reduce the performance. In this evaluation, the performance of bigrams and trigrams dropped by 1%–13% in comparison with the unigrams. The reduction in accuracy may be due to the data sparsity problems during the training of classifier using higher *n*-values ([Farhoodi et al. 2011](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)).

**Effect of Removing Topmost Common Frequent Words on Classification Performance**

Several words are domain specific and occur frequently in the entire corpus. These words appear with high frequency in both classes, and thus tend to have less discriminating power toward the text categorization. For example, the words “road, highway, lane” are very frequent in a contract document of a transportation project, and subsequently the presence of these words cannot indicate a statement to be a requirement statement. Therefore, the effect of removing such low-content words on the classification performance was analyzed. The process of removal of the common frequent words included the following two steps. First, the words appearing in the requirement and nonrequirement text were listed separately in the descending order according to their frequencies. Following this, the common words in top *k*-words of both lists were discarded from the training corpus in each experiment. The threshold value of *k* was gradually increased from 5 to 50 with an interval of 5. The common frequent words found in the first 10 experiments are shown in Table [3](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg).

Table 3. Common frequent words in requirements and nonrequirements text

| **Threshold value *k*** | |
| --- | --- |
|  | **Common frequent words in requirements and nonrequirements** | |
| 5 | design | |
| 10 | design | |
| 15 | road, design | |
| 20 | schedule, road, design | |
| 25 | project, management, schedule, design, section, construction, road | |
| 30 | project, management, traffic, schedule, mean, section, design, construction, road, system | |
| 35 | project, ministry, highway, management, plan, traffic, schedule, mean, section, design, construction, road, system | |
| 40 | project, ministry, highway, safety, management, plan, traffic, schedule, mean, section, design, construction, road, system | |
| 45 | project, ministry, highway, safety, management, plan, traffic, schedule, mean, section, design, construction, road, system | |
| 50 | project, ministry, infrastructure, highway, safety, management, plan, traffic, schedule, new, reference, mean, section, design, construction, road, system | |

The classification results were recorded for each threshold value which revealed that the removal of domain-specific common frequent words reduced the accuracy of classification models. This shows that the words appearing with high frequency in both classes also carry discriminative power and information, which are helpful for the text categorization. The graphs shown in Fig. [6](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg) indicate the variation in the accuracy of each classifier for first 10 attempts as we increase the threshold value for frequent words from 0 to 50. A reduction of 2.23%–3.35% in classification accuracy was observed upon removing the common words in the 50 topmost frequent words of both classes. SVM experienced the maximum reduction in accuracy from 98.51% to 95.16%, which is 3.35%. Similarly, NB, FNN, LR, and Word2vec+LR experienced a reduction in accuracy equal to 2.61%, 2.60%, 2.23%, and 2.60%, respectively. Therefore, removal of common frequent domain-specific words from the corpus is not effective to increase the classification accuracy.

**Effect of Selected Parts-of-Speech (POS) Phrases on Classification Performance**

Several experiments were conducted using the selected unigram features in the dataset. Features selection was based on the types of POS tags. Each word in the corpus belongs to a specific POS family and contains considerable contextual information. The experimentation was aimed at determining the most relevant POS tagged features for this domain specific text classification problem. In order to compare the performance of different POS phrases, the main dataset was split into three subsets comprising the selected features: Dataset 1 contained the noun phrases only, Dataset 2 included the verb phrases only, and adjectives were categorized in Dataset 3. The detailed description of POS-based datasets is shown in Table [4](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg) while the sample statements included in the datasets are illustrated in Fig. [7](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg). The classifiers were trained and tested using the three new datasets to evaluate the performance of the three types of features.

As shown in Fig. [8](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg), the noun and verb features revealed a comparable performance with a maximum difference of 0.9% in classification accuracy. However, the performance dropped by 7%–12% for adjectives. The results of the first three classifiers (NB, SVM, and FNN) exposed verbs as the most discriminative features for the contractual text classification. Nouns found the second place while adjectives ranked third. In contrast, LR discovered noun phrases as more relevant than verb and adjective phrases. In terms of algorithms, SVM performed the best for the Dataset 1 and Dataset 2 with an accuracy of 95.0% and 95.9%, respectively, whereas FNN achieved the highest accuracy of 84.4% for Dataset 3. The other POS tagged phrases were not considered in this study as they did not carry significant useful information. In the literature, a study conducted by Xia et al. ([2011](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)) found adjectives as the most important features for sentiment analysis problem with the verbs and nouns ranked second and third, respectively. However, this trend is not found in our results, which may be due to different dataset and domain. Therefore, it can be inferred that the most discriminative POS tagged features primarily depend on the classification algorithm and the dataset used for training and testing of the classifier.

**Practical Implications of the Study**

In the realm of practice, this study highlights several important implications for various stakeholders including construction bidders, contract administrators, and dispute resolution professionals. Construction bidders often face challenges in reading the lengthy contract documents in the short bidding period ([Lee et al. 2019](https://ascelibrary.org/doi/full/10.1061/%28ASCE%29LA.1943-4170.0000379?casa_token=_L9OCyA_lKUAAAAA%3AG6kXBqqRRhe3G9tHNpUiyJpOhxd3rhmn--JlmSZtdBj2kCKvLIT-3y7Jeg_33wktDVLMwcJYz0SAfg)). The implementation of an automated requirement extraction model can significantly reduce the reading time of professionals. The time saved in this way can be utilized in performing other technical tasks such as the preparation of best technical and financial estimates to win the project. Moreover, the implementation of automated model shall prevent the missing of any critical requirements in the contract. Precise extraction of all explicit and implicit requirements can help contract administrators to identify any objectionable requirement and the involved risks before signing the contract. Early identification of such undesirable requirements can prevent the construction firms from a significant financial loss. Improving the accuracy of contractual requirements extraction can further avoid the disputes among the contracting parties during the execution of the project. However, if the dispute arises, the model can save the time of dispute resolution professional in identifying the accurate requirements in the contract. The collective findings of the study provide support to enable friendly working environment throughout the project by ensuring precise comprehension of the client requirements at the early stage.

**Conclusion**

Project requirements are the obligations described in the contract documents; thus, the accurate identification and extraction of these requirements are crucial for the precise scope comprehension to deliver a successful project with complete satisfaction of client. The current practices of requirement extraction by manually reviewing the contract documents are tedious and time-consuming. The study has developed an automated method to extract all explicit and implicit requirements from the contract documents to support easy tracking of these requirements in the execution phase. Different classification methods are used, and their performances are compared in classifying the contractual text into two predefined categories, which include requirements and nonrequirements.

The classification models were trained and tested on a dataset of 1,787 statements extracted from the construction contract documents. A simple rule-based classification was performed at first, which achieved an accuracy of 90.6%. To improve the performance, four different machine learning algorithms were implemented, and their performances were assessed under different settings of preprocessing techniques with unigram features. All the classification models achieved promising accuracies of over 95%; however, the best performance in terms of classification accuracy was reported by SVM model, which consistently performed the best in all experiments. Among all other performance measures, the recall value was the most important in this study as it showed if the model was missing any requirement statement. SVM achieved the highest recall of 95%, which is closer to the ideal value of 100%. FNN and LR also resulted in recall values higher than 90%.

Furthermore, the study revealed the unigram model as the best classification model in comparison with other *n*-gram models. The reduction in performance may be due to data sparsity issues in classifier training using long sequences of words. Also, the effect of removing common frequent domain-specific words from the dataset on classification accuracy of unigram models was analyzed. The experiments showed a reduction in accuracy as we increased the threshold value of top frequent words considered from both classes. This reduction in accuracy indicates that the common frequent words also carry the discriminative power so removing such words from the corpus may reduce the performance. Moreover, the evaluation using selected POS tagged phrases in the dataset revealed that verbs as the most discriminative features for text classification. However, logistic regression showed different results and found nouns as the most important features for text classification. Therefore, it can be deduced that the most discriminative POS tagged phrases primarily depends on the classification algorithm and the dataset.

In addition, an experimental study was conducted to validate the effectiveness of requirement detection model by comparing the human and machine performance. The model yielded better results than humans in terms of time as well as recall. The model achieved a promising recall value of 98.15% in comparison to the human performance of 49.98% for the speed group and 53.44% for the baseline group. In terms of time, the model consumed 1.88 s to extract the requirement statements from a nine-page document in comparison to the 15 min of speed group and 30 min of baseline group. These promising results prove the adequacy of the model for the requirement extraction task.

**Data Availability Statement**

The training dataset and the classification models generated or used during the study are available from the corresponding author by request.