


Article

A Rule-Based Approach to Embedding Techniques for Text Document Classification

Asmaa M. Aubaid ^{1,2} and Alok Mishra ^{1,3,*} 

¹ Department of Modeling and Design of Engineering Systems (MODES), Department of Software Engineering, Atılım University, Ankara 06830, Turkey; asmaamuhamed1971@yahoo.com

² Ministry of Higher Education and Scientific Research/Science and Technology, Directorate: Information Technology, Baghdad/Al-Jadriya 10070, Iraq

³ Faculty of Logistics, Molde University College (Specialized University in Logistics), 6410 Molde, Norway

* Correspondence: alok.mishra@himolde.no or alok.mishra@atilim.edu.tr

Received: 11 April 2020; Accepted: 5 June 2020; Published: 10 June 2020



Abstract: With the growth of online information and sudden expansion in the number of electronic documents provided on websites and in electronic libraries, there is difficulty in categorizing text documents. Therefore, a rule-based approach is a solution to this problem; the purpose of this study is to classify documents by using a rule-based. This paper deals with the rule-based approach with the embedding technique for a document to vector (doc2vec) files. An experiment was performed on two data sets Reuters-21578 and the 20 Newsgroups to classify the top ten categories of these data sets by using a document to vector rule-based (D2vecRule). Finally, this method provided us a good classification result according to the F-measures and implementation time metrics. In conclusion, it was observed that our algorithm document to vector rule-based (D2vecRule) was good when compared with other algorithms such as JRip, One R, and ZeroR applied to the same Reuters-21578 dataset.

Keywords: text classification; rule-based; word embedding; Doc2vec

1. Introduction

There has been an urgency in terms of classifying the information available online in the past ten years, prompting researchers to focus on automatic text classification (ATC). A widely used research method for this problem depends on rule-based and embedding techniques. In the 1960s, rule-based approaches began to emerge; however, they became more common in the 1970s and 1980s [1]. The late 1980s witnessed the formation of same-time or concurrent operations and activation of rules within production systems, all of which carried on into the following decade. The rule-based system includes a set of rules that can be implemented for many purposes, including the support of decision-making or for a predictive decision in real implementations. It is possible to divide the methods of creating rules into the categories of ‘conquer and separate’ [2] and ‘divide and conquer’ [3]. This produces categorization rules in the intermediate form of a decision tree, such as C4.5, C5.0, and ID3 [2]. In the same manner, ID3 is a covering technique [4] with an approach in the form of ‘if then’ rules. The structure of the systems of rule-based methods depends on logic-specific types, such as deterministic logic, fuzzy logic, and probabilistic logic. It can also divide the system of rule-based into the following types: systems of fuzzy rule-based, deterministic rule-based and probabilistic rule-based and determine rule-based systems being in the context of bases of the rule, which includes bases of modular rules, single rules, and chained rules [5]. In practice, the task of ensemble learning may be performed in a parallel form, in a distributed manner or on mobile platforms according to given computing environments. Finally, rule-based systems are divided into three types, namely distributed, mobile and parallel [6]. The Reuters-21578 newswire benchmark and 20 Newsgroups are the most widely used benchmark

corpora in the research community in the text categorization field. They can be found in comparative studies of different approaches using flat (i.e., non-hierarchical) category systems in this corpus [7]. Hierarchical text classifiers are among the first works in this field, experiments with two classifiers on the subset of the Reuters collection reported by Kollar and Sahami [8]. Our rule-based and embedding models contributed to classifying the categories of Reuter’s dataset such as (Acq, Corn, Crude, Earn, Grain, and Ship) according to their contents. The objective of this manuscript is to provide deeper information about the performance of embedding to rule-based text classification. The main research question to explore is how varying rule-based affects the performance of text classification and we investigate the performance differences when combining our rules-based and one of the embedding models such as (doc2vec) in the task of text classification on the different datasets. We will implement many steps to acquire robust rules-based and embedding models using the Reuter 21578 and 20 Newsgroup corpora so as to make text categorization (TC) easier. Finally, Figure 1 summarizes the essential steps for a rule-based approach.

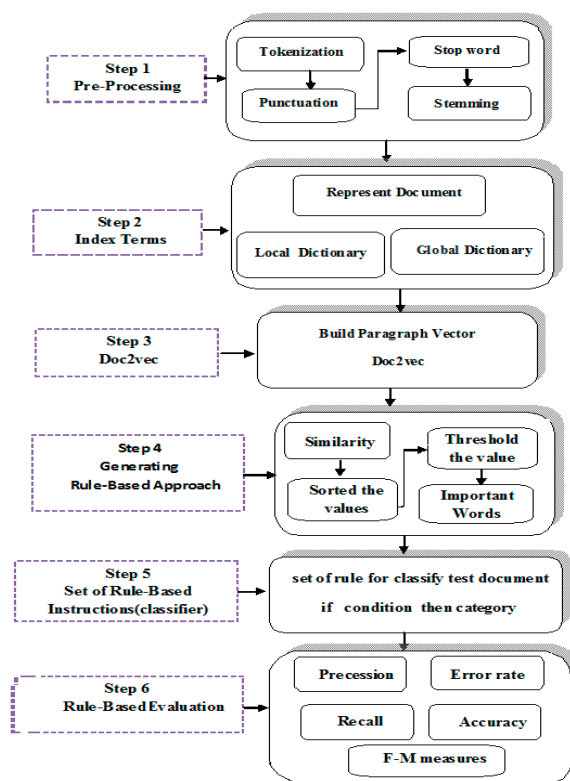


Figure 1. Steps of rule-based processing.

2. Related Work

Various studies relating to classification have been carried out taking a number of approaches. In classification systems, a rule-based learning approach to text categorization is utilized. Imaichi and Yanase suggested using rule-based methods selectively depending on the nature of the information to be extracted and make comparisons with the machine learning [9]. The model of rule-based learning consists of a set of rules learned from data [10]. Han Liu introduced an integrated framework for the design of systems of rule-based to implement missions of categorization, which included the process of rule representation, rule generation, and rule simplification. The study stressed the importance of the combination of different types of algorithms of rule learning techniques via ensemble learning [5]. Rule-based machine translation (RBMT) considers the unclear points pertaining to morphology and lexicon as serious challenges. A contribution by Rios and Göhring [11] describes an approach to resolving the forms of the morphologically ambiguous verb if a rule-based decision is not possible

due to tagging errors or parsing. Cronin et al. developed an automated patient portal message classifiers with the rule-based approach using natural language processing (NLP) and the bag of words [12]. Ganglberger et al. discuss different automatic spike detection methods in order to improve detection performance and establish a user-adjustable sensitivity parameter mainly by examining the functioning of a rule-based system, artificial neural networks (ANNs) and random forests [13]. Accordingly, the rule-based system needed a feature selection to classify text documents. Feature selection can be performed by following one of three approaches, i.e., filter, wrapper or embedded approaches [14]. In this study, this depends on embedded methods to select a feature. The optimal parameters are learned by using the embedded method to perform the feature selection approach [15]. INRA (Institut national de la recherche agronomique) and Cnrs (Centre national de la recherche scientifique) at University Paris Saclay proposed a two-step method to normalize multi-word terms with concepts from a domain-specific ontology. In this method, they used vector representations of terms computed with word embedding information and hierarchical information from ontology concepts [16]. Le and Mikolov presented word2vec and later introduced the doc2vec algorithm based on adjusted techniques for learning how to embed texts identical to word2vec, thus turning doc2vec into a branch of word2vec [17]. In their work, doc2vec was applied to model embedding for text categorization. The motivation of this study was to classify text documents by taking a rule-based approach to embedding techniques and this work will assist us in determining the acceptable methods to follow for in-text categorization based upon measuring the related criteria. Finally, this manuscript comprises eight sections containing all the necessary information related to the rule-based approach using automatic text classification for the top ten categories of the Reuters-21578 and 20 Newsgroups data sets. Then, the paper is structured as follows: Section 1 introduces the rule-based and text classification approaches. Section 2 presents related work with the study. Section 3 explains in detail the research methodology. Section 4 presents the data analysis and results. Section 5 discusses the study results. Finally, Sections 6–8 provide a conclusion, future research directions and limitations

3. Materials and Methods

3.1. Embedding Methods

The presently applied rule-based with embedding technique comprises numerous factors and is used in many applications, one being text categorization. Embedding is one of the promising applications of unsupervised learning as well as transfer learning because embedding is induced by the large unlabeled corpus. Embedding is used two character-level embedding models (fast text and ELMo) and two document-level models (doc2vec and InferSent) to compare with word-level word2vec, all in accordance to the novel approach introduced by References [18]. The rule classification system uses the doc2vec model which is a type of the document embeddings of one of the embedding methods. There are three types of embedding: word2vec, character and doc2vec, as shown in the following.

3.1.1. Word Embedding

We can define “a word embedding” as content representation such that words of similar meaning also receive the same representation. This method deals with representing documents and words and it may be seen as one of the keys in the procedures ahead of deep learning when testing natural language processing (NLP) problems. Furthermore, it is a category of methodologies in which vectors that are real-valued are used in a predefined vector space to represent single words. Every word is determined to be a vector and the values of the vectors are discovered following a neural network method. Later, the technique is usually grouped into a profound learning field. The key to the approach is to use a densely dispersed representation for each word. A real-valued vector is utilized to represent each word, frequently tens of, or many, measurements. This is divided into hundreds and thousands or matched to larger numbers of dimensions required to represent a word, such as one-hot encoding [19].

3.1.2. Character Embedding

Word2vec is arranged based on the character n grams in a character embedding model. As character n grams is shared across words assuming a closed-world alphabet, these models can generate embedding for out of vocabulary (OOV) words as well as words that occur infrequently. The two character-level embedding patterns of fastText may be used as those appearing in References [20,21], which describe ELMo in the following manner:

- fastText: applies a 300 dimensional model pre-trained on Common Crawl and Wikipedia via the Continuous Bag of Words (CBOW). To generate a representation for joint multiword expressions (MWE), fastText considers every word as whitespace delimited, taking away every space and handling them in the form of a united compound. For instance, 'couch potato' becomes 'couchpotato.' In the case of paraphrases, it uses the same word averaging technique as word2vec.
- ELMo: utilizes the Elmo Embedder group of Python's allennlp library, being pre trained in SNLI and SQuAD, with a dimensionality of 1024. It is noted that the essential use case for ELMo is implemented by generating embedding in context. However, it does not provide any context in the input for compatibility with the other models. Thus, the benefits of the full potential of this model are unknown. Therefore, ELMo is not suitable since the relative compositionality of a compound is often predictable from its component words only [18], so the present study makes use of doc2vec.

3.1.3. Document Embedding

In this study, doc2vec is a proposal for paragraph-level embedding from the research team responsible for word2vec. It is possible to use the doc2vec approach to learn a model that can create an embedding technique in a specific document. In contrast to some of the known used methods (such as averaging word vectors, n gram models and bag of words (BOW)), doc2vec is public too and it can be utilized to create embedding from any length of text. From large corpora of raw text, it can train doc2vec in a totally unsupervised fashion. Doc2vec operates effectively once applied to represent extended texts [22]. In this paper, doc2vec (Distributed Memory Model) is used. Doc2vec is an offset to the present word embedding models and it is a popular method to learn word vectors. Moreover, it can be divided into two partitions.

3.1.3.1. A Distributed Memory Model

This part of doc2vec contributes to our study; our methodology for learning doc2vec is a model inspired by the techniques to learn the vectors of words. The inspiration is to provide a commitment to a forecast about words following in the sentence. Although the fact that a word vector is instated arbitrarily, as an indirect result, it can capture semantics from the forecasting task. Therefore, it will use this idea in our doc2vec and in the identical method. The doc2vec can also help in the estimating task of the following word since there are numerous settings that were inspected from the paragraph. In our doc2vec framework (see Figure 2), a unique vector is mapped to every paragraph and described by a column in matrix D for every word W. The two-word vectors and doc2vec are concatenated or averaged to estimate the next word in a context. In the experiments, the concatenation method was used to consolidate the vectors. The section token can be thought of as another word, and the paragraph works as memory recalling what is absent from the subject or the setting of the section itself. For the previous reason, the Distributed Memory Model of Paragraph Vectors (PV DM) is named doc2vec. The contexts are fixed-length and are inspected from a sliding window over a section. The paragraph of a vector is shared over all contexts produced from the same paragraph, but not over the paragraphs. In this model, predicting the fourth word is possible by using the chain or the average of the related vector along with a context of three words. The doc2vec is assumed to be the absent data from the present context and it can function as a memory of the paragraph subject. After being trained, the doc2vec can be utilized as vocabularies for the paragraph. In summary, the algorithm has two main stages:

1. Training stage to acquire word vectors W , soft max weights U ; b and doc2vec D on as of now observed paragraphs.
2. The inference stage to acquire doc2vec D for new paragraphs by including more columns in D and a gradient descending on D while holding W , U , and b fixed. D is utilized to make a prediction about various specific labels utilizing a standard classifier [17].

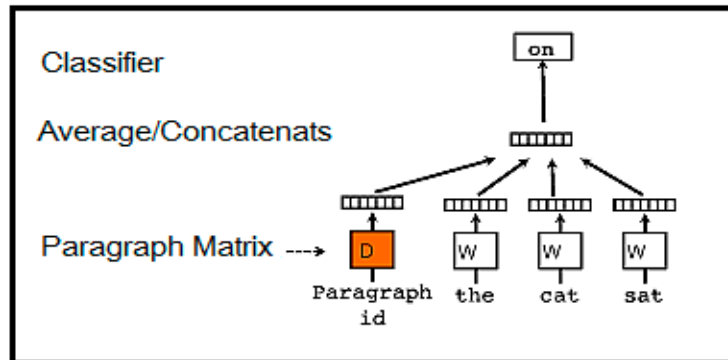


Figure 2. Learning model for doc2vec illustrating information absent in the present context and acting as a memory of the paragraph subject [17].

3.1.3.2. Distributed Bag of Words Model

The method described earlier involves forecasting the following words in a text window by using a concatenation of the doc2vec by vectors. Another approach is to eliminate the context words at the source by driving the model to estimate words sampled in any manner from the paragraph in the output. In reality, it means that for each round of stochastic gradient descent, a text window is examined, followed by sampling a random word from the text window and forming a classification task given the doc2vec, as shown in Figure 3 [17].

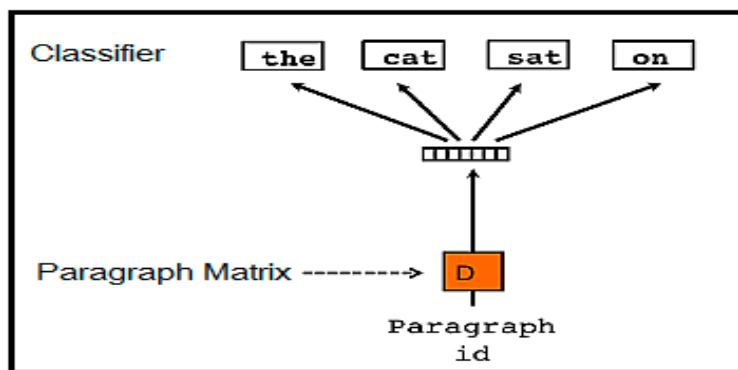


Figure 3. Version of doc2vec with Distributed Bag of Words. Here, the doc2vec was trained to predict the words in a small window [17].

To build the doc2vec model, the major stages of training need to be prepared and tested with a dataset, as shown in the following.

3.2. Data Sets

3.2.1. Data Sets Types

A dataset is defined as a collection of related, but separated, features of related data that can be accessed individually or in combination. It can be formed and arranged into a type of data structure. For example, a dataset may be contained in the collection of business data (identity, salaries, names,

address, contact information, etc.). It is possible to transform a database and use it as a set of data and we can connect the data inside it with a particular type of information. We will evaluate the doc2vec model for our rule-based approach applied to the following two datasets.

3.2.1.1. Reuters-21578 Dataset

The Reuters Newswire in 1987 saw the emergence of documents of the Reuters-21578 collection and this is a publically available version of the well-known Reuters-21578 “ApteMod” corpus for text categorization. A Reuters Ltd. (S. Weinstein, S. Dobbins, and M. Topliss) and the Carnegie Group, Inc. (M. Cellio, P. Andersen, P. Hayes, Ir. Nirenburg, and L. Knecht) collected and indexed these documents according to certain categories. The Reuters-21578 collection is distributed in 22 files. Each of the first 21 files (reut2-000.sgm through reut2-020.sgm) contains 1000 documents, while the last (reut2-021.sgm) contains 578 documents. The files are in SGML format. Rather than going into the details of the SGML language, it is described how the SGML tags are used to divide each file, and each document, into sections. Each of the 22 files begins with a document type declaration line: `<!DOCTYPE lewis SYSTEM “lewis.DTD”>`The documents of Reuters-21578 are divided into training and test sets. Each document has five-category tags, namely, TOPICS, PLACES, PEOPLE, ORGS, and EXCHANGES. Each category has the number of topics that are used for a document, but in this study focuses on the TOPIC category only.

3.2.1.2. 20 Newsgroups Dataset

About 20,000 newsgroup documents are collected in the 20 Newsgroups dataset and these documents are partitioned (almost) equally into 20 separate newsgroups. The data are organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (including newsgroups such as comp.sys.mac.hardware and comp.sys.ibm.pc.hardware), while others are highly unrelated (including newsgroups such as misc.forsale and soc.religion.christian) [23].

3.2.2. Pre-Processing Data Sets

Pre-Processing is an important step for initializing the text, it takes an amount of processing time. Pre-Processing includes several steps such as tokenization, punctuation, stop and stop words.

3.2.2.1. Tokenizing

Tokenizing is a process of cutting the input text into pieces of words/tokens by remembering the sequence in the text that is in the tokenization and simultaneously discarding specific characters, such as punctuation [24]. Tokenizing is defined as the process of breaking down documents into words or terms called tokens. An entire text is lowercased when all the punctuation is removed and when applying the process of tokenization [25].

3.2.2.2. Punctuation

Defined as a set of marks, they are used to make sentences flow smoothly and express meaning accurately. These marks determine the place of pause or provide a signifying feeling to our words. Punctuation makes sentences pure by breaking ideas. Moreover, the punctuation points quotes, out titles and other main parts of the language. Finally, punctuation is vital in any text, necessitating their introduction. Examples include “,” “!” “?” “*”.

3.2.2.3. Stop Word Removal

One important step in text classification is to eliminate stop words. A stop word is defined as a list of commonly used words that have an important function in a text but no meaning. A stop word in a text is removed to reduce noise terms, and as a result, the keyword remains [26]. Stop words are

common words occurring in most documents, such as “the,” “and,” “from,” “are,” “to,” etc. They are required to apply this processing because these stop words cannot decide the category of the document in the categorization system [25].

3.2.2.4. Stemming

When acquiring information, stemming changes a word form to its root by means of specific principles related to the target language [27]. This is vital due to the presence of affixes, which consist of prefixes, infixes, suffixes, and confixes (combinations of prefixes and suffixes) in derived words [28]. Stemming is a process of reducing the terms to their roots. For example, words such as “working,” “worker” and “worked” are reduced to “work” and “crumbling” and “crumbled” are reduced to “crumb.” This process is used to reduce the computing time and space as different forms of words are stemmed into a single word. In fact, this is the main advantage of this process [25].

3.3. Local Dictionary Creation

It is the role of the main dictionary to perform feature selection in text categorization with a different set of features being selected from each category. Several studies have been conducted which used this type of dictionary. In the local dictionary, a contrasting set of features is selected from each, independent of the other categories, and that dictionary works to increase the speed of the classification process for each category by selecting the most important features in that category. Table 1 introduces the local dictionary for a number of categories in the dataset.

Table 1. Local dictionary for some categories of the dataset.

Acq	Corn	Crude	Earn	Grain	Ship
dollar	govern	rise	second	govern	govern
unit	soybean	govern	unit	new	spokesman
relate	total	spokesman	rate	hectare	new
hold	office	new	bank	china	strike
place	report	study	project	credit	office
commissar	maize	total	bus	soybean	japan
share	ton	unit	result	total	report
result	union	end	figure	unit	iran
debt	earth	tanker	debt	end	boat

3.4. Rule-Based Approach

The rule based approach is considered to be one of the most flexible methods by which the black box of the process of the text classification technique can be shown. The details of a process of classification can be observed and it can add a number of tools or new instructions to obtain good results. The next subsections will explain the approach of rule-based in briefly.

3.4.1. Rules—Preliminary

A rule-based system is commonly comprised of a set of if then rules [29] expressed such that there are various approaches to information representation in the area of artificial intelligence. However, the most famous one may be in the form of if then rules defined as: “IF cause (antecedent) THEN effect (consequent).”

1. Rules: Data are used to derive the most known symbolic representations of knowledge:
 - A natural and smooth form of representation → possible search by humans and their interpretations;
2. A standard form of the rules;
3. If “condition” then “class”;
4. Other forms: Class if conditions; conditions → class.

3.4.2. Rules—Formal Notations

Rule-based processes, also known as expert or generation systems represent a type of artificial intelligence. The rules in this system are used as the learning representation for the information that is coded into the system [30]. The expert system affects the implications of rule-based systems completely and it copies the reasoning of human experts in explaining an information-intensive issue. Instead of learning in a declarative, static manner as a course action of things that are valid, rule-based systems can be considered to be knowledge that can be represented as a set of rules determining what to do or what to conclude in various situations.

3.4.3. Structure of a Rule-Based Expert System

In the early 1970s, Simon and Newell from the University of Carnegie Mellon proposed a production system model which is the foundation of modern rule-based expert systems [31]. The idea of that production model was based on whenever humans applied knowledge (expressed as production rules); they can solve any problem represented by problem-specific information. The problem-specific information or facts in short-term memory and the production rules were stored in long-term memory. A rule-based expert system has five components: the database, the knowledge base, the explanation facilities, the inference engine, and the user interface [32].

3.5. Classification Methods

Classification is a data mining technique that assigns items in a set to the target class. The aim of classification is to visualize correctly the target categories for each case in the data [33]. Three rule-based classification methods are applied in addition to our rule-based (D2VRule) method that is taken as a benchmarking algorithm to be studied for the Reuters-21578 and 20 Newsgroups datasets.

3.5.1. JRip (RIPPER)

This algorithm is one of the essential and most well-known. A set of rules in growing the size is used to examine classes and a premier set of rules for each category is created using JRip (RIPPER) with gradually reduced errors by handling all the instances of a special decision in the training data as categories. It returns a set of rules that cover every member of that class. Therefore, it proceeds to the next categories and does the same, repeating these processes until every category covered [34].

3.5.2. One Rule (OneR)

Abbreviated to OneR, this method uses a simple algorithm in a text classification technique to create a decision tree with one level. From different instances, OneR can deduce simple but precise classification rules. In spite of its simplicity, OneR is able to treat lost values and lost numeric attributes more flexibly. The OneR algorithm generates one rule for each predictor (class) in the data. The rule with the minimum false rate is selected by depending on the principle of one rule for each attribute in the training data [35].

3.5.3. ZeroR

ZeroR is considered to be the simplest classification method based on the target and it disregards all other predictors. In spite of ZeroR lacking predictability power, it is helpful in determining the performance of a baseline as a metric for other classification methods. ZeroR constructs a hesitancy table for the feature and selects its highest hesitancy value [36].

3.6. Evaluation Measures

The performance evaluation feature selection approaches are computed using recall (R) and precision (P) [37].

3.6.1. Precision

Precision is defined as a percentage of relevant documents correctly retrieved by the system having a symbol (TP) with respect to every document relevant to humans ($TP + FN$) [37]:

$$Precision = \frac{TP}{TP + FP} = \frac{Retrieved\ Relevant}{Retrieved}, \quad (1)$$

where

- TP (true positive) is defined as the correctly assigned number of documents to Class (i).
- FP (false positive) is defined as the incorrectly assigned number of documents to Class (i) by the classifier but which actually does not belong to that class.

3.6.2. Recall

The percentage of relevant documents correctly retrieved by the system (TP) with respect to every document relevant to humans is $TP + FN$. In other words, recall is equal to the ratio of the retrieved relevant documents to the relevant documents [37].

$$Recall(R) = \frac{TP}{TP + FN} = \frac{Retrieved\ Relevant}{Relevant}, \quad (2)$$

where:

- TN (true negative) is defined as the classifier not assigning documents to Class (i); they actually do not belong to Class (i).
- FN (false negative) is defined as the classifier not assigning documents to Class (i); however, they actually do belong to Class (i).

3.6.3. F-Measure

This element is defined as a global estimation of the performance of an information retrieval (IR) system by combining measure precision (P) and recall (R) in a single measure called F-measure [37].

$$F - M = \frac{2(TP)}{FP + FN + 2(TP)} = \frac{2(Retrieved\ Relevant)}{Relevant + Retrieved + 2(Retrieved\ Relevant)}. \quad (3)$$

3.6.4. Error Rate Inverse of Accuracy

This element is defined as a global estimation of the performance of an information retrieval (IR) system by combining measure precision (P) and recall (R) in a single measure called F-measure [37].

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN}. \quad (4)$$

3.6.5. Accuracy

Accuracy is defined as the percentage of documents correctly classified by the system [36]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \quad (5)$$

4. Experiment Setup

In this section, we describe the experimental setup of a text classification system which includes the preprocessing, documents representation, rule-based induction in addition to evaluation metrics.

4.1. Preparing the Dataset

A collection of documents was used from the Reuters-21578 dataset (for the training dataset) and the top ten categories were selected for the 20 Newsgroups dataset. The first step of a prepared dataset was implemented using:

1. Tokenization
2. Punctuation
3. Stop word
4. Stemming

These approaches are explained in detail in the previous sections. Figure 4 explains the steps of preparing a dataset for a rule-based approach.

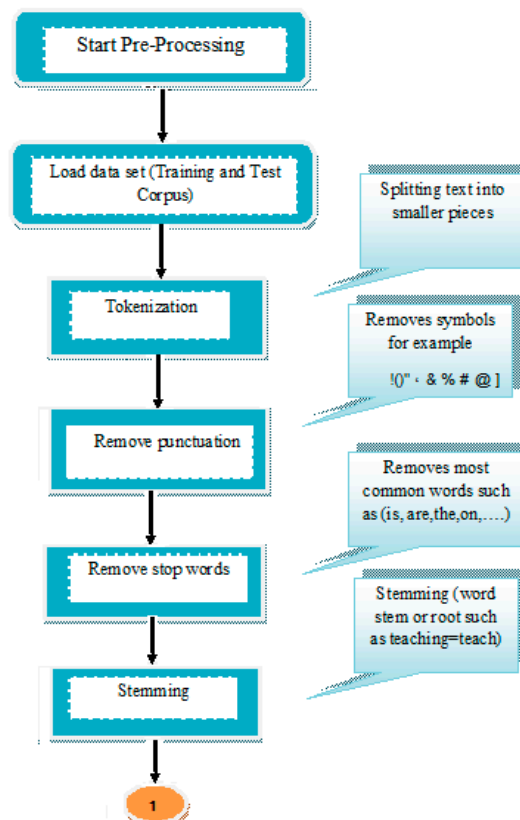


Figure 4. Dataset preparation for a rule-based approach.

4.2. Rule-Based Processing (Documents Representation)

The previous steps were necessary to begin the rule-based creation process; however, the following sections are more important to build our rule-based by using the doc2vec approach under a titled document to the vector rule based (D2VRule).

4.2.1. Terms Indexing

The term indexing was considered a necessary step to build the dictionary and had benefits for classification processes. This dictionary was named a local dictionary; it was considered the main dictionary to apply feature selection in text categorization. In this dictionary, a different set of features was selected from each category. Several studies have been performed using the local dictionary policy. In the local dictionary, a contrasting set of features was selected from each independently of the other categories, and this dictionary helped to increase the ability of the classification process for each category by selecting the most important terms in that category.

4.2.2. Doc2vec Creation

In this step, the doc2vec approach was taken (explained in detail in previous sections). A doc2vec model was built by using documents of the training dataset. This step was necessary in order to determine the similarity between vocabularies, which were sets of familiar words in the language of a document of a local dictionary as well as training documents to acquire important features selection (vocabularies). These were used to classify text documents of the test corpus.

4.2.3. Computing Similarity of Vocabularies

The vocabulary was extracted from documents by training the data set such that words that were similar or had a related meaning to other words were extracted. This can be of benefit when one wishes to avoid repeating the same word by concentrating on the value of similarity of vocabulary near to 1 and removing the vocabulary which has a value near to 0 (zero) by depending on a threshold value. The procedure of similarity was performed by building a doc2vec model to prepare documents and compute the similarity of vocabularies in a local dictionary with doc2vec itself using special instructions in Python (most similarity).

4.2.4. Sorting of Vocabularies

The values of similarities of vocabularies were arranged according to threshold values defined as points beyond which there was a change in the manner a program executes. In particular, the threshold value was represented as the value of the similarity of terms in documents and by which it determined the important words in these documents. Figure 5 presents the steps used to implement the rule.

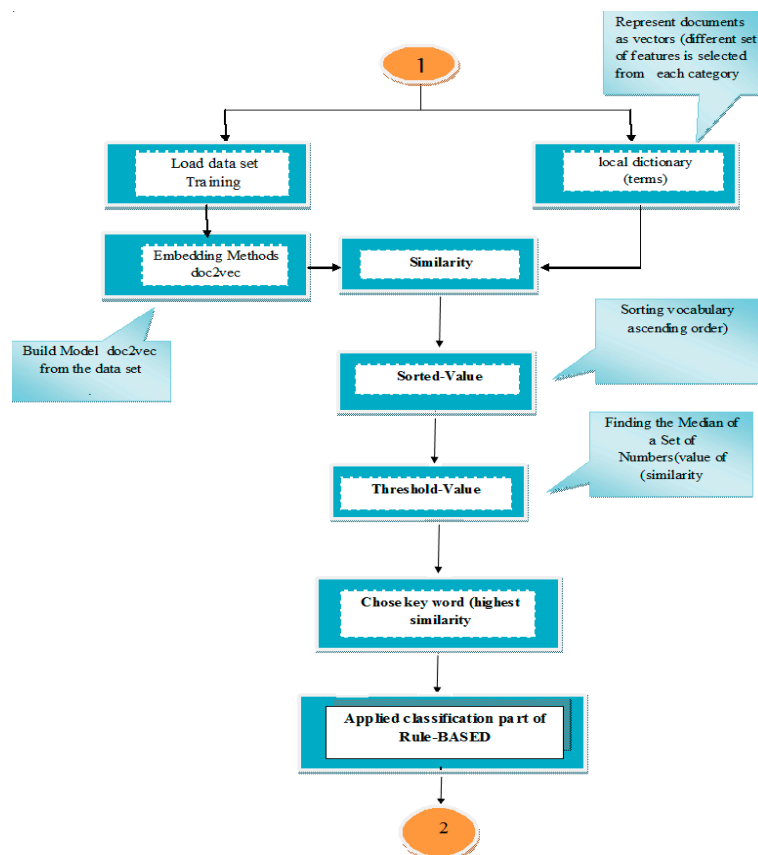


Figure 5. Steps for starting applied classification with rule-based (D2VRule).

4.3. Rule-Based Induction

Promising results can be obtained when applying our rule-based (D2VRule) to the number of standard problems in the text classification. Therefore, to classify the objects, it is necessary for most learning algorithms in the first step to transform these objects into a representation suitable for concept learning. The transformation process of electronic texts is discussed in the previous section of Part 1 and Part 2. In the D2VRule, as in other rule induction systems, it is defined as a decision rule that is a set of Boolean clauses linked by logical (AND, OR) operators which together imply membership in a particular class. A sequence of rules ending in a default rule with an empty set of clauses usually builds a hypothesis of a classification. When we apply the classification process, it can divide the core of the rules-base into two parts, with the left-hand sides of the rules being implemented sequentially until one of them evaluates to true, and the right-hand side of the rule being offered as the class prediction.

4.3.1. Set of Rule-Based Instructions

Text categorization was implemented by depending on a measurement metric called the feature selection metric. Its general idea was to determine the importance of words (vocabularies) using a measure that can remove non-informative words and retain informative words.

The following are some of the rule-based structures that will be generated for one of ten categories:

- if (“corn” \in doc or “maiz” \in doc)
- or (“wheat” \in doc and “maiz” \in doc)
- or (“tonn” \in doc and “wheat” \in doc
- and “corn” \in doc))then
- category = “corn”

4.3.2. Rule-Based Evaluation

The rule-based categories were checked according to dataset categories and classification rules, followed by evaluation measurements being computed. The evaluation measurements include:

1. Precision measurements
2. Recall measurements
3. F-Measures
4. Error rate
5. Accuracy

Examples of an induction rule and the evaluation metrics are shown in Figure 6.

Finally, it can arrange the pre-processing of our rule-based approach according to the block diagrams in Figure 7 and build the block diagram of the rule-based technique. The following figure shows the processing of the rule-based approach for two partitions, which was used in the text classification technique.

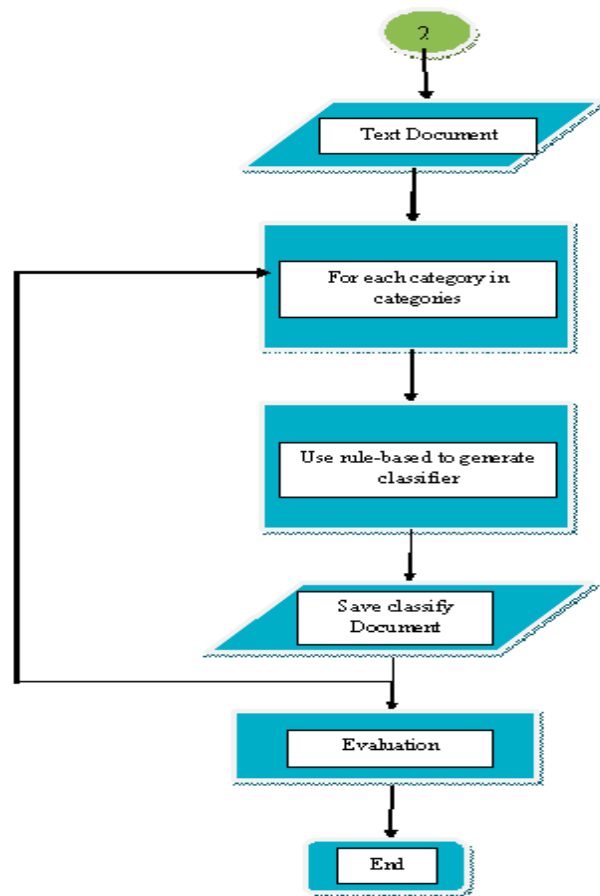


Figure 6. Steps of classification of rule-based and evaluation metrics.

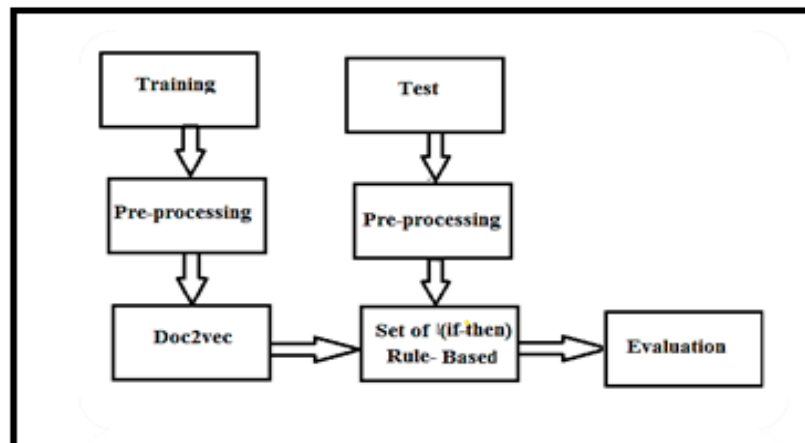


Figure 7. Rule-based approach.

5. Results

In this section, it is possible to encounter extensive investigations of precision, recall, F-measure, error rate, and accuracy criteria. Moreover, precision and recall formulations (Equations (1)–(4)) were used for the Reuters-21578 and 20 Newsgroups datasets to classify the top ten categories individually. The computations were compared in order to select the acceptable method to implement the text classification. In addition, our rule-based approach examined the acq, corn, crude, earn, grain, interest, money-fix, ship, trade, and wheat top ten categories for the Reuters-21578 dataset. For the 20 Newsgroups dataset, our rule-based approach examined the categories of data sets.

As seen in Figures 8–12, we explored the precision, recall, F-measures, error rates, and accuracy of a rule-based approach to classify the test documents when we selected the top ten categories of the Reuters-21578 and 20 Newsgroups datasets.

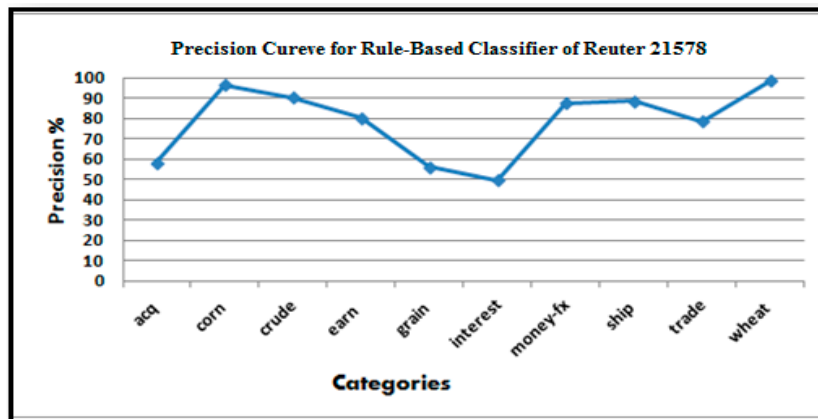


Figure 8. Variation in the precision of the rule-based text categorization technique for the Reuters-21578 dataset.

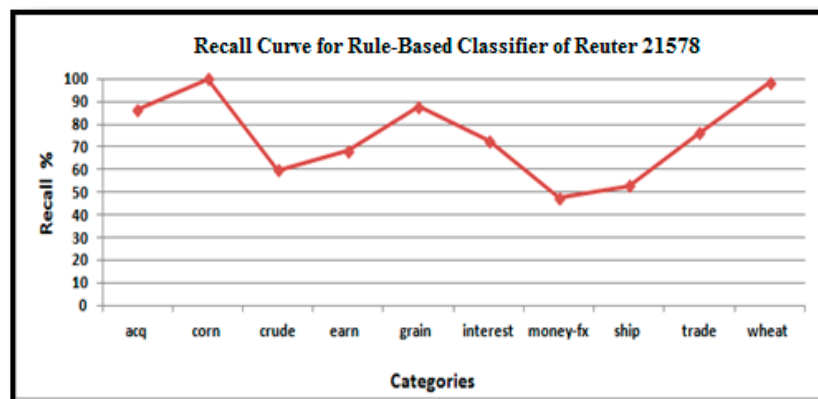


Figure 9. Variation in the recall of the rule-based text categorization technique for the Reuters 21587 dataset.

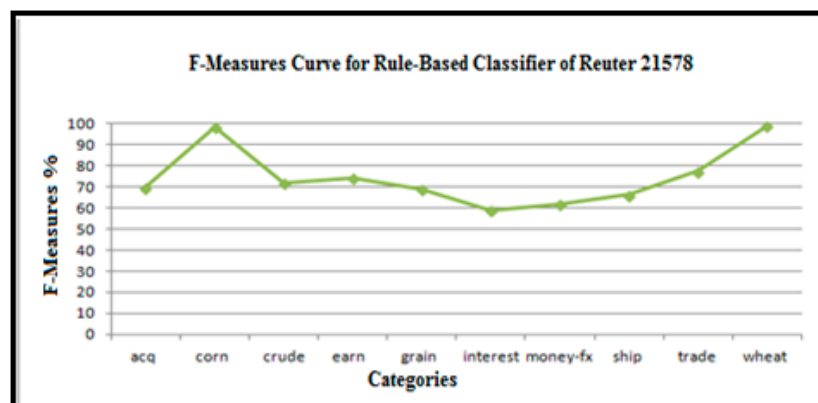


Figure 10. Variation in F-measures of the rule-based text categorization technique for the Reuters-21578.

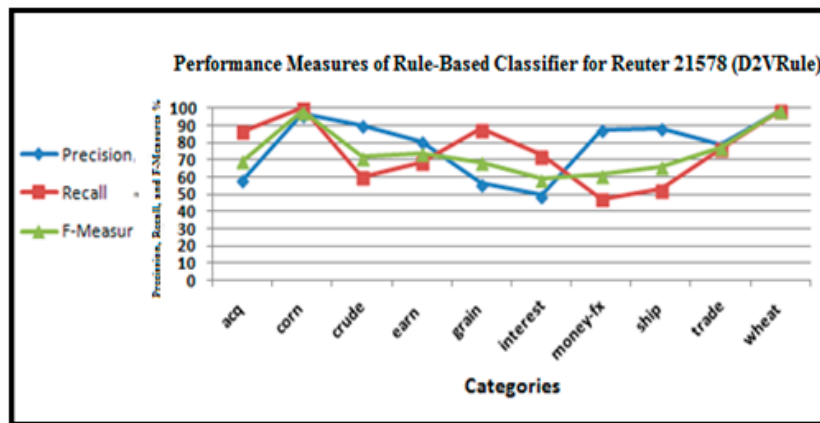


Figure 11. Performance measures of the rule-based classifier for the Reuters-21578 dataset.

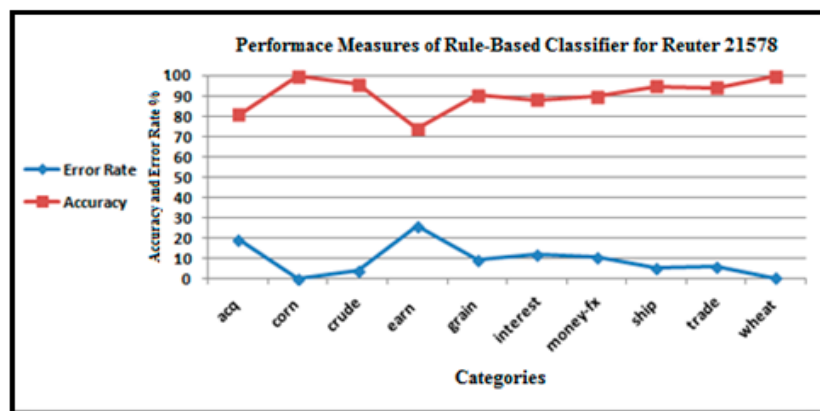


Figure 12. Performance of the rule-based text categorization technique for the Reuters-21578 dataset.

As shown in Figures 13–18, we explored the precision, recall, and accuracy of a rule-based approach to classify the test documents when we selected the top ten categories of the 20 Newsgroup dataset.

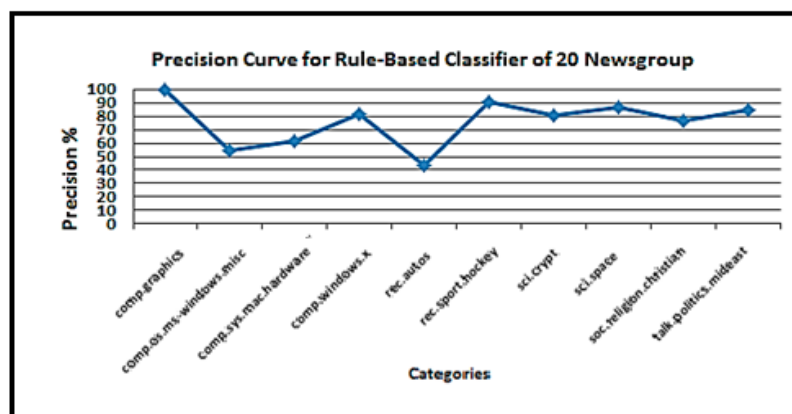


Figure 13. Variation in the precision of the rule-based text categorization technique for the 20 Newsgroups dataset.

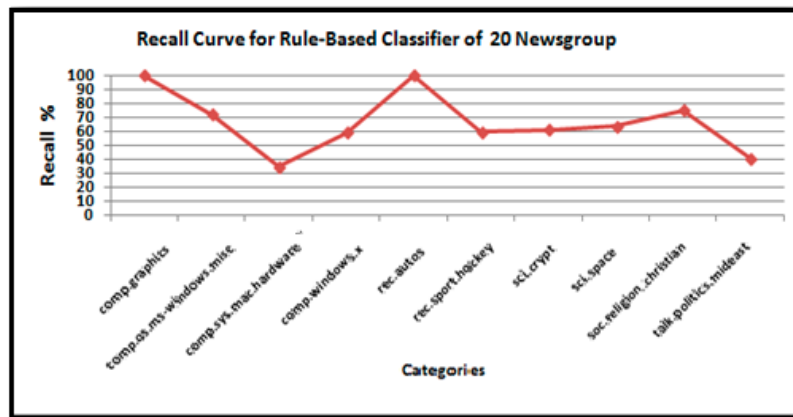


Figure 14. Variation in the recall of the rule-based text categorization technique for the 20 Newsgroups dataset.

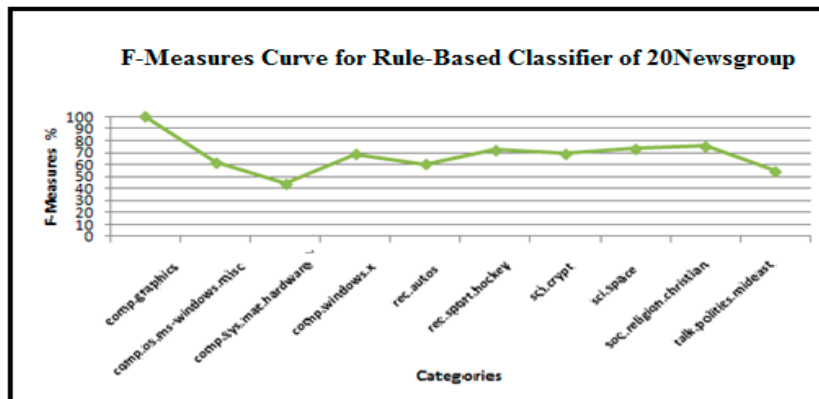


Figure 15. Variation in the F-measures of the rule-based text categorization technique for the 20 Newsgroups dataset.

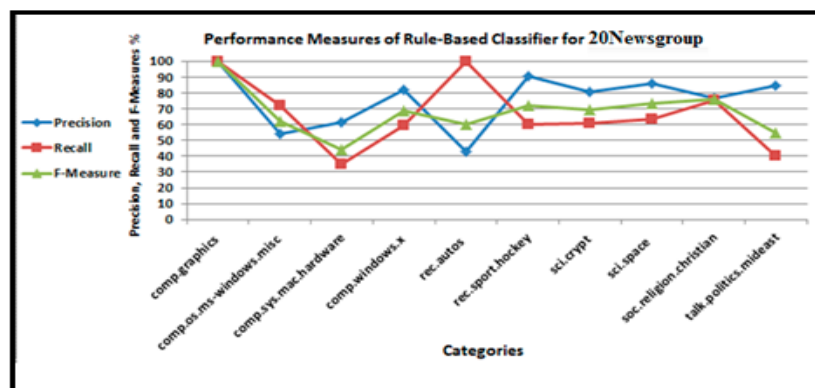


Figure 16. Performance measures of the rule-based classifier for the 20 Newsgroups dataset.

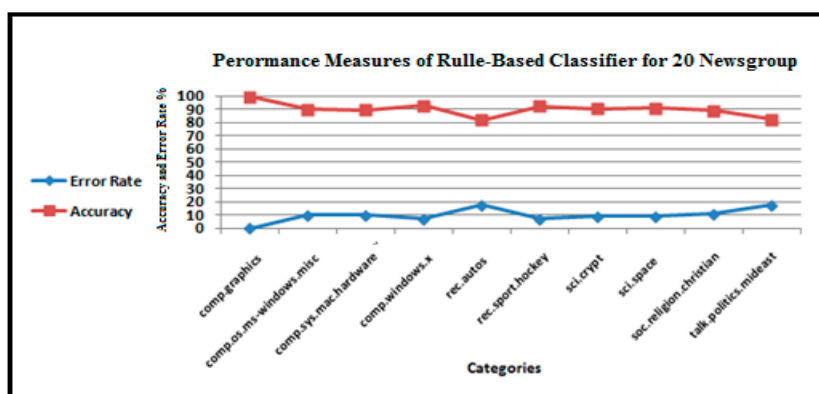


Figure 17. Performance of the text categorization system technique for the 20 Newsgroups dataset.

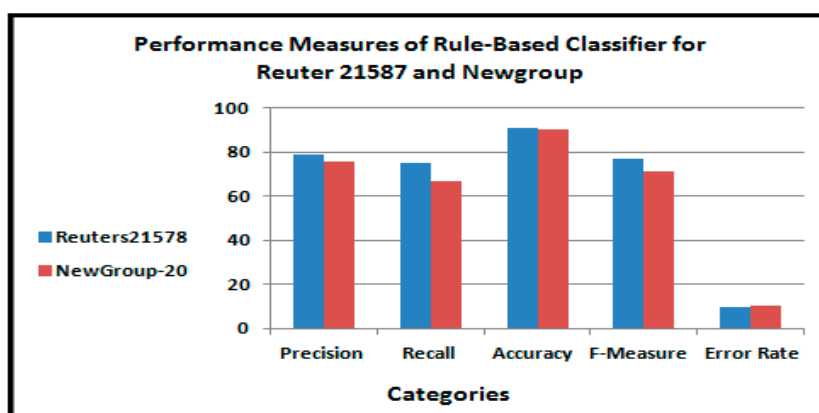


Figure 18. Comparison of performance results using the rule-based text categorization technique applied to different categories within the Reuters-21578 and 20 Newsgroups datasets.

Finally, when the rules for JRip, OneR and ZeroR were applied to the Reuters-21578 dataset, we obtained F-measures and accuracy metrics of 0.713–0.752, 0.506–0.598 and 0.219–0.39 for JRip, OneR and ZeroR, respectively. Table 2 introduces the comparison measurements among three rule-based classification methods, and the precision and recall of the system were averaged by using the micro-average method.

Table 2. Shows the comparison of the metrics among JRip, OneR, and ZeroR on the Reuters-21578 dataset.

It	Parameters	D2VRule	JRip	OneR	ZeroR
1	Precision	79	70.4	48.4	15.2
2	Recall	75	75.9	59.8	39.0
3	F-Measure	76.75	71.3	50.6	21.9
4	Accuracy	90.72	75.2	59.8	39

6. Discussion

The development of computer technologies, rule-based techniques, and automatic learning techniques can make information retrieval technology easier and more efficient. There exist many approaches to decision-making, such as rule-based and artificial neural networks. The rule-based approach is considered one of the most flexible methods by which the black box of the process of text classification techniques can be shown. The details of a process of classification can be seen and it can add some tools or new instructions to obtain good results. All preprocessing on two datasets (Reuters-21578 and 20 Newsgroups) is implemented using the Python programming language, an open-source tools framework, and a document-level embedding (doc2vec) technique to represent

text documents being used, which appears to be more effective in the preparation of data. In addition, the rule-based approach would support the classification approach by improving the recall, precision and accuracy measurements of classification.

A suitable vocabulary (informative words) is selected according to the following criteria:

1. The highest value of similarity of the feature
2. The highest numbers of the term frequency (numbers of repetitions of important words in documents)
3. Highest numbers of document frequency (number of documents including the feature)

The recall, precision, F-measures, error rate and accuracy are obtained according to a suitable choice of vocabulary selection. It is clear that there are precision and other metrics evaluations in a rule-based approach to classify categories of test datasets affected by the above criteria. According to Ligeza [29], symbolic rules are some of the most popular knowledge representation and reasoning methods. Therefore, we have many reasons to view the rule-based approach as superior to other approaches. Firstly, for the naturalness of expression, expert knowledge can be used as guiding rules. Secondly, we have modularity, in which the rules-based approach can be considered an independent method. Thirdly, the restriction of syntax allows the construction of rules and checking of consistency using other programs [38]. Fourthly, it is a compact representation of general knowledge; it can easily form the representation of general knowledge about a problem. Fifthly, the provision of explanations is represented by the ability of the rule-based approach to provide explanations for any derived conclusions in a direct manner, which is considered to be a vital feature [39]. The information extraction techniques of the rule-based approach have been used effectively in commercial systems and are favored because they are easily understood and controlled [40]. The rule-based approach and temporal specificity score TSS based classification approaches are proposed, and the results show that the proposed rule-based classifier outperforms the other four algorithms by achieving 82% accuracy, whereas the TSS classification achieves 77% accuracy [41]. In 2019, Li et al. [42], they proposed a model where the performance for it was still good and mostly stable with respect to the F-measure, and from the curve of this measure, when the number of extracted keywords N was 7, the F-measure reached a maximum of 43.1% compared to Xia's work. [43], in which, the basic idea of TextRank used for keyword extraction was introduced. The process of constructing candidate keywords and the F-measure up to 37.28%, and all these previous results were less than our results, where F-measures reached 76.75%. Decision table, Ridor, OneR, DTNB and PART are five algorithms applied to the chess end game dataset and by using evaluation metrics to check the performance of these algorithms, it appears from the results that PART is the accept rule-based classification algorithm when compared with other studied rule based algorithms.

On the other hand, the OneR algorithm showed an overall low performance for every parameter, and when these results were matched with our results when applying the OneR rule to the Reuters-21578 dataset, it became clear that the OneR algorithm had low values of precision, recall, and F-Measures [44]. A single attribute-based classification (SAC) is needed to divide the original dataset into multiple one-dimensional datasets. The experimental results show that SACs performed better than the classical OneR algorithm. The performance of different classification methods was examined on the large dataset [45].

The algorithms tested were SMO, J48, Zero, OneR, RPart and the Naive Bayes algorithm. It was discovered that the highest error was found in the ZeroR classifier with an average score of approximately 0.5. The other algorithms ranged on average 0.1–0.2. Therefore, the ZeroR technique is not a good option for the classification of any dataset due to its many errors [46], these results are in agreement with our conclusion. The performance of three rule classifier algorithms, namely RIDOR, JRIP and Decision Table, using the Iris datasets, was calculated using the cross-validation parameter. Finally, it was observed that the JRIP technique is not a good option for classification [47], and when applying those algorithms to our dataset, it became apparent that our algorithm agrees

with the results of other algorithms. In Reference [48], an improved hierarchical clustering algorithm has been developed based on association rules and these algorithms were tested on benchmark data set Reuters-21578, and the results (F-measures) produced by the association rule-based hierarchical clustering (ARBHC) method are better than the results of the traditional hierarchical algorithm, and these results (F-M equal to 29%) are so much less than our results. uRule is a new rule-based classification and prediction algorithm, it was proposed to classify a limited number of uncertain data, and the accuracy of the uRule classifier remains relatively stable like our rule-based, but our rule was applied on a huge of documents within Reuters and 20Newsgroups datasets [49]. Reference [50] presents a new technique using state-of-art machine-learning methods, deep learning, and it is used to solve the problem of choosing the best structures and architectures of neural networks out of many possibilities, and it introduced the RMDL (random multimodel deep learning) for classification that combines multi deep learning models to produce better performance, they have evaluated this approach on datasets such as the Web of Science (WOS), Reuters, MNIST, CIFAR, IMDB, and 20NewsGroups dataset, Furthermore, the proposed approach shows improvement in classification accuracy for both text and image classification. Finally, this accuracy for Reuters-21578 and 20Newsgroups datasets in the best case is equal to 90.69% and 87.91% respectively, but our result related to accuracy for the same datasets was 90.72%, 90.07% respectively.

This provides a better classification process according to evaluation metrics.

7. Conclusions

We selected our rule-based approach to classify text documents into ten categories for two datasets, which in our case were the Reuters-21578 and 20 Newsgroups datasets. Computer programming using Python was implemented. It was expected that these results would be beneficial for information retrieval systems and this work has assisted us in setting the acceptable methods for use in text classification by depending on precision, recall and accuracy approaches. In conclusion, the results were, in the case of Reuters-21578, 79% precision, 75% recall, and 76.75% F-measures, 9.28% error rate, and 90.72% accuracy measurement. For the 20 Newsgroups dataset, the results were 76% precision, 66.64% recall, 70.98% F-measures, 9.93% error rate, and 90.07% accuracy measurement. When we compared our algorithms with other algorithms (JRip, OneR, and ZeroR) for the Reuters-21578 dataset and by using the performance factors of precision, recall, F-measure, error rate, and classification accuracy, it was observed that our algorithm performed better than other algorithms and had a good classification process. Our intention is to make some improvements to the rule-based approach so as to be more active with the real-time dataset of the Reuters agency as well as selecting new types of machine learning.

8. Future Work

We intend to make further contributions, with some enhancements to the rule-based approach, which are more active with real-time datasets, such as newspaper datasets. Tagging content or products using categories as a way to improve browsing or to identify related content on your website. Platforms such as E-commerce, news agencies, content curators, blogs, directories, and likes can use automated technologies to classify and tag content and products.

9. The Limitations

Text classification is an important research problem in many fields. However, there are several challenges remaining in the processing of textual data [51].

1. Our results pertain to two specific datasets, namely Reuters-21578 and 20 Newsgroups.
2. We worked to improve the classification technique by taking a large number of documents in the training part of the dataset since the volume of the training data had an important role in learning a model. Training data must be labeled and be large enough to cover all upcoming classes.

3. The rules used were only for the English language and by adding a number of modifications; they can be suitable for a new language.
4. Information retrieval systems experience the diverse nature of texts with highly variable content, quality and length.

Author Contributions: Conceptualization, A.M.A. and A.M.; Data curation, A.M.A.; Formal analysis, A.M.A.; Investigation, A.M.; Methodology, A.M.A. and A.M.; Software, A.M.A. and A.M.; Supervision, A.M.; Validation, A.M.A.; Visualization, A.M.A.; Writing—original draft, A.M.A.; Writing—review and editing, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Levy, O.; Goldberg, Y. Linguistic regularities in sparse and explicit word representations. In Proceedings of the Eighteenth Conference on Computational Language Learning, Baltimore, MD, USA, 26–27 June 2014; pp. 171–180.
2. Quinlan, R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1993; ISBN 1 55860 2380.
3. Partridge, D.; Hussain, K.M. *Knowledge Based Information Systems*; McGraw Hill, Inc.: New York, NY, USA, 1994; ISBN 0077076249.
4. Michalski, R.S. On the quasi-minimal solution of the general covering problem. In Proceedings of the Fifth International Symposium on Information Processing, Bled, Yugoslavia, 8–11 October 1969; Volume A3, pp. 125–128.
5. Han, L.; Alexander, G.; Cocea, M. *Rule Based Systems for Big Data: A Machine Learning Approach*; Springer: Cham, Switzerland, 2015; ISBN 10 3319236954.
6. Mendel, J.M. *Uncertain Rule Based Fuzzy Systems*; University of Southern California: Los Angeles, CA, USA, 2017; ISBN 978 3 319 51369 0.
7. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv.* **2002**, *34*, 1–47. [[CrossRef](#)]
8. Koller, D.; Sahami, M. Hierarchically classifying documents using a very few words. In Proceedings of the International Conference on Machine Learning, San Mateo, CA, USA, August 1997; Morgan Kaufmann: San Mateo, CA, USA, 1997; Volume 14, pp. 170–178.
9. Imaichi, O.; Yanase, T.; Niwa, Y. A Comparison of Rule Based and Machine Learning Methods for Medical Information Extraction. In Proceedings of the International Joint Conference on Natural Language Processing Workshop on Natural Language Processing for Medical and Healthcare Fields, Nagoya, Japan, 14–18 October 2013; pp. 38–42.
10. Panthong, R.; Srivihok, A. Wrapper feature subset selection for dimension reduction based on ensemble learning algorithm. *Procedia Comput. Sci.* **2015**, *72*, 162–169. [[CrossRef](#)]
11. Rios, A.; Göhring, A. Machine Learning Applied to Rule Based Machine Translation. In *Hybrid Approaches to Machine Translation*; Costa Jussà, M., Rapp, R., Lambert, P., Eberle, K., Banchs, R., Babych, B., Eds.; Theory and Applications of Natural Language Processing; Springer: Cham, Switzerland, 2016.
12. Cronin, R.M.; Joshua, D.F.; Denny, J.C. A Comparison of Rule Based and Machine Learning Approaches for Classifying Patient Portal Messages. *Int. J. Med. Inform.* **2017**, *105*, 110–120. [[CrossRef](#)] [[PubMed](#)]
13. Ganglberger, W.; Gritsch, G.; Hartmann, M.M.; Fürbass, F.; Kluge, T. A Comparison of Rule Based and Machine Learning Methods for Classification of Spikes in EEG. *JCM* **2017**, *12*, 589–595. [[CrossRef](#)]
14. Suto, J.; Oniga, S.; Sitar, P.P. Comparison of wrapper and filter feature selection algorithms on human activity recognition. In Proceedings of the 2016 6th International Conference on Computers Communications and Control (ICCCC), Oradea, Romania, 10–14 May 2016; pp. 124–129.
15. Naseriparsa, M.; Bidgoli, A.M.; Varae, T. A hybrid feature selection method to improve the performance of a group of classification algorithms. *Int. J. Comput. Appl.* **2013**, *69*, 28–35. [[CrossRef](#)]

16. Ferré, A.; Deléger, L.; Zweigenbaum, P.; Nédellec, C. Combining Rule Based and Embedding Based Approaches to Normalize Textual Entities with an Ontology. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; pp. 3443–3447.
17. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 24 June 2014; Volume 32.
18. Nandakumar, N.; Salehi, B.; Baldwin, T. A Comparative Study of Embedding Models in Predicting the Compositionality of Multiword Expressions. In Proceedings of the Australasian Language Technology Association Workshop, Dunedin, New Zealand, 10–12 December 2018; pp. 71–76.
19. Stefanowski, J. Induction of Rules. Ph.D Thesis, Institute of Computing Sciences, Poznan University of Technology, Catania Troina, Italy, April 2008.
20. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [[CrossRef](#)]
21. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M. Deep contextualized word representations. In Proceedings of the NAACL-HLT, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
22. Lau, J.H.; Baldwin, T. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proceedings of the 1st Workshop on Representation Learning for NLP. Association for Computational Linguistics, Berlin, Germany, 11 August 2016; pp. 78–86.
23. Rennie, J. 20 Newsgroup Data Set. Available online: <http://qwone.com/~jason/20Newsgroups/> (accessed on 5 March 2019).
24. Deshmukh, P.R.; Ade, R. Classification of Students Using Psychometric Tests with the Help of Incremental Naive Bayes Algorithm. *Int. J. Comput. Appl.* **2014**, *89*, 27–31.
25. Karamcheti, A.C. A Comparative Study on Text Categorization. M.Sc. Thesis, University of Nevada Las Vegas, Las Vegas, NV, USA, 2010.
26. Arthana, R. Stop Word Indonesian (and Implementation on Apache LUCENE). Available online: <http://www.rey1024.com/2012/06/stop-word-bahasa-indonesia-dan-implementasi-pada-apache-lucene/> (accessed on 10 June 2019).
27. Asian, J.; Williams, H.E.; Tahaghoghi, S.M.M. *Stemming Indonesian*; ACM: New York, NY, USA, 2003; Volume 6, pp. 1–33.
28. Adriani, M.; Asian, J.; Nazief, B.; Tahaghoghi, S.M.M.; Williams, H.E. Confix Stripping: Approach to Stemming Algorithm for Indonesian Language. *ACM Trans. Asian Lang. Inf. Process.* **2007**, *6*, 13.1–13.33. [[CrossRef](#)]
29. Ross, T.J. *Fuzzy Logic with Engineering Applications*; John Wiley & Sons Ltd.: West Sussex, UK, 2004.
30. Ligeza, A. Logical Foundations for Rule based Systems. In *Studies in Computational Intelligence*, 2nd ed.; Springer: Heidelberg, Germany; AGH University of Science and Technology Press: Kraków, Poland, 2006; pp. XX, 309. ISBN1 10 3540291172. ISBN2 13 9783540291176.
31. Simon, A.; Newell, A. Human Problem Solving: The State of the Theory in 1970. *Am. Psychol. Assoc.* **1970**, *26*, 145–159. [[CrossRef](#)]
32. Negnevitsky, M. Rule based expert systems. In *Artificial Intelligence, a Guide to Intelligent Systems*, 2nd ed.; Pearson Education Limited: Harlow, UK, 2005; pp. 3–5.
33. Andreeva, P.; Dimitrova, M.; Radeva, P. Data Mining Learning Models and Algorithms for Medical Application. In Proceedings of the 18-th Conference on Systems for Automation of Engineering and Research SAER, Sofia, Bulgaria, April 2004; pp. 11–18.
34. Rajput, A.; Aharwal, R.P.; Dubey, M.; Saxena, S.P. J48 and JRIP Rules for E Governance Data. *IJCSS* **2011**, *5*, 201–207.
35. Buddhinath, G.; Derry, D. A Simple Enhancement to One Rule Classification. Ph.D. Thesis, Department of Computer Science & Software Engineering University of Melbourne, Melbourne, Australia, 2006.
36. Available online: <https://www.saedsayad.com/zeror.htm> (accessed on 23 March 2019).
37. Manning, C.D.; Prabhakar, R.; Hinrich, S. *Introduction to Information Retrieval*; Cambridge University Press: New York, NY, USA, May 2008; ISBN 13: 978 0521865715.
38. Available online: <https://www.merospark.com/> (accessed on 13 April 2019).

39. Prentzas, J.; Hatzilygeroudisa, I. Categorizing Approaches Combining Rule Based and Case Based Reasoning. In *University of Patras and Technological Educational Institute of Lamia Greece*; Wiley online library; Blackwell Publishing Ltd.: Hoboken, NJ, USA, April 2007; Volume 24, Issue 2, pp. 97–122.
40. Khademi, S.; Haghighi, P.D.; Burstein, F.; Palmer, C. Enhancing rule based text classification of neurosurgical notes using filtered feature weight vectors. In *Australian Conference on Information Systems*; University of Wollongong, Computer Science: Wollongong, Australian, 2016; pp. 1–11.
41. Khan, S.U.R.; Islam, M.A.; Aleem, M.; Iqbal, M.A. Temporal specificity based text classification for information retrieval. *Turk. J. Electr. Eng. Comput. Sci.* **2018**, *26*, 2915–2926. [[CrossRef](#)]
42. Li, J.; Huang, G.; Fan, C.; Sun, Z.; Zhu, H. Key word extraction for short text via word2vec, doc2vec, and textrank. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1794–1805. [[CrossRef](#)]
43. Wen, Y.; Yuan, H.; Zhang, P. Research on Keyword Extraction Based on Word2Vec Weighted TextRank. In *Proceedings of the 2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China, 14–17 October 2016; pp. 2109–2113.
44. Mahajan, A.; Ganpati, A. Performance Evaluation of Rule Based Classification Algorithms. *Int. J. Adv. Res. Comput. Eng. Technol.* **2014**, *3*, 3546–3550.
45. Du, L.; Song, Q. A Simple Classifier based on a Single Attribute. In *Proceedings of the 14th International Conference on High Performance Computing and Communications*, Liverpool, UK, 25–27 June 2012; pp. 660–665.
46. Nasa, C. Evaluation of Different Classification Techniques for WEB Data. *Int. J. Comput. Appl.* **2012**, *52*, 34–40. [[CrossRef](#)]
47. Veeralakshmi, V.; Ramyachitra, D. Ripple down Rule learner (RIDOR) Classifier for IRIS Dataset. *Int. J. Comput. Sci. Eng.* **2015**, *4*, 79–85.
48. Rose, D.J. An Effect Association Rule-Based Hierarchical Algorithm for Text Clustting. *Int. J. Adv. Eng. Technol.* **2016**, *751*, 753.
49. Qin, B.; Xia, Y.; Prabhakar, S.; Tu, Y. A Rule-Based Classification Algorithm for Uncertain Data. In *Proceedings of the IEEE 25th International Conference on Data Engineering*, Shanghai, China, 29 March–2 April 2009; pp. 1633–1640. [[CrossRef](#)]
50. Heidarysafa, M.; Kowsari, K.; Brown, D.E.; Meimandi, K.J.; Barnes, L.E. An Improvement of Data Classification using Random Multimodel Deep Learning (RMDL). *Int. J. Mach. Learn. Comput.* **2018**, *8*, 298–310.
51. Losiewicz, P.; Oard, D.W.; Kostoff, R.N. Textual Data Mining to Support Science and Systems. *J. Intell. Inf. Syst.* **2000**, *15*, 99–119. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).