Rich document representation and classification: An analysis

Mostafa Keikha\textsuperscript{a,}\textsuperscript{*}, Ahmad Khonsari\textsuperscript{d}, Farhad Oroumchian\textsuperscript{b}\textsuperscript{b}

\textsuperscript{a}University of Tehran, Department of Electrical and Computer Engineering, Gorgan 49139-66883, Iran
\textsuperscript{b}University of Wollongong in Dubai, The College of Informatics and Computer Science, United Arab Emirates

\textbf{ARTICLE INFO}

\textbf{A B S T R A C T}

There are three factors involved in text classification. These are classification model, similarity measure and document representation model. In this paper, we will focus on document representation and demonstrate that the choice of document representation has a profound impact on the quality of the classifier.

In our experiments, we have used the centroid-based text classifier, which is a simple and robust text classification scheme. We will compare four different types of document representations: N-grams, single terms, phrases and RDR which is a logic-based document representation. The N-gram representation is a string-based representation with no linguistic processing. The Single term approach is based on words with minimum linguistic processing. The phrase approach is based on linguistically formed phrases and single words. The RDR is based on linguistic processing and representing documents as a set of logical predicates. We have experimented with many text collections and we have obtained similar results. Here, we base our arguments on experiments conducted on Reuters-21578. We show that RDR, the more complex representation, produces more effective classifier on Reuters-21578, followed by the phrase approach.

\textbf{1. Introduction}

Text classification is the task of assigning one or more classes to a passage from a predefined set of classes and it can be done in different ways. The distribution of the class members among training and test dataset is an important factor in determining success of the classification algorithm.

Most of the classification algorithms utilize the concept of the distance between two documents, which is computed based on the difference between the values of the features sets representing the documents. The choice of the feature set has a profound effect on the quality of the classification. Different document representation methods create different types of feature sets.

There are many different document representation models. The simplest is N-gram where words are represented as strings of length N. The most popular and effective representation is single words, where documents are represented by their words. Most often the stem of the words is used instead of the words themselves. Stemming the words increases the possibility of matches between the documents and the queries or other documents. A little more sophisticated approach involves extracting statistical or linguistic phrases and representing documents with their stemmed single words and phrases. All the above approaches assume term independence, i.e., relevance of one term does not provide any clues for the relevance of the other terms.

There are document representation models like Logical Imaging\textsuperscript{[12]} that do not assume term independence and are able to represent term relationships. Some other systems like PLIR\textsuperscript{[13]} use logic to represent the relationships among the words in the text. PLIR uses NLP-based methods to discover the relationship among different words or phrases in the collection. Then it uses a logical form to represent a document. This logical representation is called RDR or Rich document representation. In this article we want to show that in the context of document classification the quality of the representation is of utmost importance. We also point out that a logical representation such as RDR which represents the documents more qualitatively performs better than the simpler document representations.

The rest of the paper is organized as follow. In Section 2, we will describe vector space model with more details. Section 3 describes our classification approach, and in Section 4 the document representation approaches will be discussed along with the distance and similarity measures. Section 5 describes the experimental results and Section 6 presents the conclusion of the obtained results.

\textbf{2. Vector space}

The vector space model is one of the most common models for representing documents and widely used in document classification.
tion. In this model, each document is represented as a vector of terms. The terms are the features that best characterize the document and can be anything from strings of length $N$, single words, phrases or any set of concepts or logical predicates. The vector space model does not keep any information regarding the order in which the terms occur. Often before processing the terms, stop-words, terms with little discriminatory power, are eliminated. Also it is common to use the stems of the words instead of the actual words themselves.

All the terms in the dataset define a “space” in which each term represents one “dimension”. For distinguishing a document from the other documents, numeric values are assigned to each term in order to show the importance of that term in that document.

The base weighting schema in vector space model uses two main factors; the frequency of a term in the document or term frequency (tf); and the inverse of the number of documents containing that term or inverse document frequency (idf). The tf factor indicates the importance of a term in document and is a document-specific statistic which is usually normalized. The idf factor is a global statistic and measures how widely a term is distributed over the collection. There are many functions for calculating tf and idf factors described in [15] and [4].

Another element in weighting schema is the normalization factor that usually tries to diminish the effect of document length in weighting. We will explain our normalization factor in the weighting subsection.

3. Text classification

There is a wide range of classification methods available for the task of text classification. SVM or support vector machines method [23] is based on a learning approach to solve the two class pattern recognition problems. The method is defined over a vector space where the problem is to find a decision surface that best separates the documents into two classes. KNN, or K nearest neighbor method, on the other hand, classifies documents according to the voting of its k nearest neighbor documents, which can be discovered using a distance measure between documents [23]. Neural Network has also been used in classification task. However, Han and Karypis showed that the two methods of SVM and kNN significantly outperform Neural Network approach when the number of the positive training instances per category is small [10].

Another method is the centroid-based text classification [10], which is a very simple yet powerful method that outperforms other methods on a wide range of datasets. In this method, given a new document, D, which is to be labeled, the system computes the similarity of D to the centroid of the existing classes. The centroid of a class is defined as the average of the document vectors of that class. The following formula shows how the centroid of a specific class is calculated:

$$C_k = \frac{\sum_{D_i \in \text{Class}_k} \bar{D}_i}{|\text{Class}_k|};$$

where $C_k$ is the $k$th centroid, and $D_i$ is the vector representation of the document $i$, and Class $k$ is the collection of documents which belong to the $k$th class.

Any document that is to be classified is compared with all of the centroids of the classes, and receives a similarity value for each of the centroids:

$$\text{Sim}(\bar{D}_{\text{new}}, \bar{C}_i) = \bar{D}_{\text{new}} \cdot \bar{C}_i$$

After comparing with all of the centroids, the system computes a score of membership for each document in each class. These scores form the category vector for that document and a threshold could then be used in calculation of the final categories. It is noticeable that working with category vectors makes the method able to deal with multiple categories for each document.

4. Document representation approaches

Document representation is the method of representing documents in a system. There are many models of representation which differ on the assumptions they make about the words and documents. The two common assumptions are term independence and document independence.

Term independence states that from relevance of one term we cannot make any statement about the relevance of other terms. Document Independence states that relevance of a document has no effect on the relevance of the other documents. The validity of these assumptions or lack of it has nothing to do with their usefulness. The simpler document representation models by accepting both of the above assumptions; treat documents as bag of terms or a set of independent features. The more complex models, may accept only one or none of the above assumptions. For example systems that use a thesaurus, only accept limited term independence where the only accepted relationships between terms are the thesaurus relations such as similarity or dissimilarity. IR systems that use clustering in grouping their output do not subscribe to document independence therefore they try to group the similar documents in to one group. Logic-based systems such as GRANT [21] do not subscribe to neither of the assumptions. The Grant system follows a semantic network approach where all the terms and documents are connected to each other through a variety of relationships.

However, the way a system uses its document representation is also an issue. Two systems with identical Pdocument representations but with different matching techniques will perform differently. For example, RDR is a logical representation which does not subscribe to either of the independence assumptions. The PLIR system utilizes RDR as a document representation model. PLIR uses reasoning in order to estimate the similarity of queries to documents or documents to each other. The power of PLIR which normally outperforms best vector space models in experimental conditions comes from both its document representation and its inferences.

In order to isolate the effect of document representation, we will apply the same weighting and matching techniques to different document representation models. For this purpose, we have selected vector space model because of its simplicity. We will demonstrate the performance of four document representation models N-grams, single words, single words plus phrases and RDR under different experimental setting.

4.1. N-grams

In the N-gram representation approach, the text is broken down into strings of $n$ consecutive characters with or without regard to word length or word boundaries [9]. Zamora uses trigram analysis for spelling error detection [11]. Damashek uses n-grams of length 5 and 6 for clustering of text by language and topic. He uses $n = 5$ for English and $n = 6$ for Japanese [17]. Some authors [7] draw n-grams from all the words in a document but use only n-grams wholly within a single word. Others [17] use N-grams that cross word boundaries, i.e., a N-gram string could start within one word and end in another word, and include the space characters that separate consecutive words.

A pure N-gram analysis does not use language-specific or semantic information. Stemming, stop-word removal, syntactically based phrase detection, thesaurus expansion, etc. are ignored by...
this method. So, theoretically, the performance of this approach should be lower than methods that make effective use of the language-specific clues. However, it is a very simple and fast method, and can be a very effective in situations where the language of the document is not previously known, or when the dataset contains textual errors. In some languages such as Persian, N-gram methods have comparable performance to that of unstemmed single words [1].

Authors of [17 and 6] show that the performance of an N-gram system is remarkably resistant to textual errors, e.g., spelling errors, typos, errors associated with optical character recognition, etc.

In our experiments, we used 3, 4 and 5-grams and did not cross the word boundaries in the creation of the N-grams. We used words without stemming and did not remove stop words.

When the n in the N-gram method increases, the method slowly loses its N-gram characteristic, and gains more of the characteristics of the single word representation method. For a special dataset such as the SIAM text mining competition dataset, we noticed that the 5 gram approach results in the best performance as explained below.

4.2. Single word representation

Single words are the most common way by which to represent documents. In this method, each document is represented as a vector of weights of its distinct words.

This method, while quite simple, is very powerful for indexing documents [16]. However, some characteristics of documents may affect the performance of this method. Spelling errors for instance, causes the incorrect weights to be assigned to words. The vector space model and the tf/idf weighting are very sensitive to weights and so in these situations, errors will result in false weights [20]. Preprocessing algorithms like error detection and correction can be helpful in these situations.

4.3. Stemmed single word representation

Stemming is a method to improve the quality of single word indexing, by grouping words that have the same stem. The most common stemming algorithm for English language is Porter algorithm [19]. This algorithm removes the common postfxes in different steps and its simplicity and high effectiveness has caused it to be used in many applications requiring stemming. In our experiments, we used this algorithm for stemming.

4.4. Phrases

Many systems index phrases along with single words. There are two ways to form phrases. First is statistical where co-occurrence information is used in some way to group together words that co-occur more than usual. Then there is syntactical approach where linguistic information is used to form the phrases [8]. For example an adjective and a noun together form a phrase. Normally the length of statistical phrases is two. But in the systems that use linguistic analysis the length of a phrase is one of the system parameters. It could be two, three or more. The statistical phrases are hard to explain to users, since they may not carry a meaning or violate familiar linguistic rules. However, syntactical phrases are more widely used in conjunction with stemmed words in order to improve the precision of the system.

4.5. Rich document representation (RDR)

Another approach in document indexing is rich document representation, in which documents are represented by a set of logical terms and statements. These logical terms and statements describe the relationships that have been found in the text with a logical notation close to multi-valued logic of Michalski. In [3] it has been reported that this kind of document indexing has improved quality of document clustering.

The process of producing these logical forms is as follows: First, the text is tagged by a Part of Speech tagger, and then a rule based extraction process is applied to the output of the part of speech tagger. Matched rules indicate the existence of special relations in the text. For example a proposition such as “for” in the sentence fragment such as “… operating systems for personal computers …” suggests a relationship between two noun phrases “operating systems” and “personal computers” [14]. Then, these relations are represented with a format similar to that of Multi-valued logic as used in the theory of human plausible reasoning i.e., operating_system (personal_computers) [2]. In a similar way, a logical statement such as operating_system (personal_computers) * (Windows_XP) can be formed from a sentence fragment such as “… Windows XP is a new operating system for personal computers …”.

Rich document representation represents documents by their stemmed single terms, stemmed phrases, logical terms and logical statements. This method provides more semantic representation for a document. PLIR system uses RDR representation and combines all the documents’ representations into a single semantic network. By doing this, the information contained in documents complete each other and creates a semantic space. PLIR applies its inferences in this semantic space to infer relevance or closeness of documents to concepts or each other [13]. The performance of this method depends on the power of the rules, and the characteristics of the dataset. Noisy text usually misleads the part of speech tagger and the pattern matching rules, and thus reduces the performance of the method. Also, the ambiguities of the natural language text, e.g. the existence of anaphora, make this method to miss some of the relations. A higher level of text preprocessing in the discourse or pragmatic level can lead to a better performance of this representation model.

5. Experiments

For evaluating effectiveness of different type of representation many sets of experiments have been performed in different collections and settings. Here, we report experiments conducted on the Reuters-21578 test collection and demonstrate that the RDR representation has better results than N-grams, single words and single words plus phrases representations. In these experiments we used centroid-based text classification as our classification method.

To divide the collection into training and test set we used Mod- Lewis split. This leads to a training set consisting of 13,625 stories and a test set consisting of 6188 stories and 135 different categories.

We created four different representations for each document including 5-gram, single term, phrase and logical representation. Using these representations, four different similarities for each document and category were computed. Here, we specified one category for each document, so the category with the highest similarity will be assigned as the category of that document.

We executed more than 90 different runs with different settings. We report on the most important ones here. The Porter stemming algorithm has been used in all of the experiments whenever stemming was required. Stop-words were also removed using Van Rijsbergen’s list for removing stop words [5]. In the weighting step, we used the same weighting schema for both training and test documents, namely Ltu.Ltu weighting schema [9], that means we used the following formulas for tf, idf and normalization factor (nf):
where term freq is the frequency of term in the document, \( N \) the number of all documents in the collection, \( n \) the number of documents that contain that term and \( \# \) of unique terms is the number of unique terms in the specified document.

Multiplication of these three parameters for each term would be the weight of that term in the specified document. We did our experiments with two values for slope (0.25 and 0.75). Table 1 shows the accuracy for each indexing method. The accuracy is defined as

\[
\text{accuracy} = \frac{TP}{TP + FP}
\]

where TP is True Positive that is the number of cases where the correct classification has been identified.

FP is False Positive that is the number of cases where the category has been incorrectly identified.

Table 1 shows the performance of each method in its simplest form. Each run uses only one type of tokens in its indexes. For example run 1 reports the performance of stemmed single words.

### Table 1

<table>
<thead>
<tr>
<th>Run. ID</th>
<th>Indexing type</th>
<th>Slope</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single</td>
<td>0.25</td>
<td>56.3</td>
</tr>
<tr>
<td>2</td>
<td>Single</td>
<td>0.75</td>
<td>56.0</td>
</tr>
<tr>
<td>3</td>
<td>Phrase</td>
<td>0.25</td>
<td>53.9</td>
</tr>
<tr>
<td>4</td>
<td>Phrase</td>
<td>0.75</td>
<td>53.9</td>
</tr>
<tr>
<td>5</td>
<td>Logic</td>
<td>0.25</td>
<td>30.9</td>
</tr>
<tr>
<td>6</td>
<td>Logic</td>
<td>0.75</td>
<td>30.9</td>
</tr>
<tr>
<td>7</td>
<td>5-Gram</td>
<td>0.25</td>
<td>47.8</td>
</tr>
<tr>
<td>8</td>
<td>5-Gram</td>
<td>0.75</td>
<td>47.8</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Run. ID</th>
<th>Indexing method</th>
<th>ID of combined runs</th>
<th>Accuracy Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>Single + Phrase</td>
<td>(1),(2)</td>
<td>59.9, 6.3</td>
</tr>
<tr>
<td>45</td>
<td>Single + Phrase + Logic (RDR)</td>
<td>(1)-(3)</td>
<td>63.4, 12.6</td>
</tr>
</tbody>
</table>

### Table 3

Some of the values used for the parameters of the OWA operator

\[
Q(r) = \begin{cases} 
0 & \text{if } r < a, \\
\frac{r - a}{b - a} & \text{if } a \leq r \leq b, \\
1 & \text{if } r > b.
\end{cases}
\]

5.1. OWA fuzzy operator

The ordered weighted averaging operator, commonly called OWA operator, provides a parameterized class of mean type aggregation operators. As part of our experiments, we have also experimented with the OWA operator to find the best combination of logical terms with single terms and phrases. Based on the formula below, the similarity of three different terms should be sorted first and then combined by using the W vector weights.

\[
similarity = \frac{w_1\cdot b_1 + w_2\cdot b_2 + w_3\cdot b_3}{w_1 + w_2 + w_3}
\]

In order to find the best possible weights we have used the Yager method [22] as described in the following formula:

\[
w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), \quad i = 1, 2, \ldots, n.
\]

In the above formula, \( n \) is the total number of all things that will be combined. Here, we are combining three similarity measures. We use the following definition for the Q function as it has been reported as one of the best in [24].

### Table 4

Some of the values used for the parameters of the OWA operator

<table>
<thead>
<tr>
<th>Run. ID</th>
<th>Indexing method</th>
<th>Weighing model</th>
<th>( W_1 )</th>
<th>( W_2 )</th>
<th>( W_3 )</th>
<th>Accuracy Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>Single + Phrase + Logic</td>
<td>Ltu</td>
<td>0.06</td>
<td>0.66</td>
<td>0.28</td>
<td>68.5, 21.6</td>
</tr>
<tr>
<td>49</td>
<td>Single + Phrase + Logic</td>
<td>Ltu</td>
<td>0.66</td>
<td>0.34</td>
<td>0</td>
<td>55.8, 0.8</td>
</tr>
<tr>
<td>50</td>
<td>Single + Phrase + Logic</td>
<td>Ltu</td>
<td>0</td>
<td>0.32</td>
<td>0.68</td>
<td>70.2, 24.6</td>
</tr>
</tbody>
</table>
The $a$, $b$, and $r$ are the parameters in this formula that could be tuned for different situations. We have experimented with different values for the above parameters including the values suggested in [22]. Table 3 shows some of the values that have worked well in our experiments.

Table 4 shows the result obtained in the experiments for the RDR document representation by using the values described in Table 3. The improvement column shows the improvement over the stemmed single words representation.

Many other experiments also have been conducted using other methods such as non-linear combination and Dempsher-Shaffer method in order to find an effective way of combining stemmed single words, phrases and logical terms in RDR document representation. It seems an accuracy rate as high as 78.9 with non-linear combination is reachable [18].

6. Conclusion

There are three factors in text categorization. Those are categorization model, similarity measure and document representation. There are many alternatives for each one of these factors. In this report, we focused on the document representation model and examined four different document representation techniques with different levels of complexity. The simplest of all is the N-gram model where the words are broken down into overlapping strings of length $N$. Most popular model is the single term approach where each word is treated as a term independent of any other term. It is common to use word stems instead of their original surface form in order to increase recall. A little more complicated model uses phrases (mostly syntactic) along with single terms. There are many more complex approaches to document representation. Here, we used a logical approach called RDR (Rich document representation) that extracts the relationships between terms and represents them as predicates. We have demonstrated that in the context of the classification, the better document representation RDR produces better classification. In these experiments RDR outperformed stemmed single-words by as much as 24% and stemmed single-words + phrases by 9%. This is not completely the same as ad-hoc information retrieval where, over the years, it has been observed that many different systems with different document representations exhibit similar performances in the context of TREC conferences.

References