Feature Selection in Document Classification

Daniel I. MORARIU, Radu G. CREŢULESCU, Macarie BREAZU

"Lucian Blaga" University of Sibiu, Engineering Faculty, Computer Science and Electrical and Electronics Engineering Department

Abstract

Text classification, as part of text mining domain, is the problem of classifying text documents into a set of predefined classes. After extracting the features, documents are represented as a vector containing a huge number of features. When working on large collections of documents both time and memory restrictions are prohibitive and, therefore, we have to apply feature selection methods to reduce the dimensionality of the vector. In this paper, we evaluate three feature selection methods: one based on the 1R classifier, Information Gain and Gain Ratio. The best results were obtained with the Information Gain method and for a relatively small dimension of the feature vector. The best accuracy of 93.58% was obtained using only 27.14% of features.

Keywords: Text Mining, feature selection, Information Gain, 1RClassifier, Gain Ratio.

1 Introduction

In the last years a significant increases in using the Web can be observed, and also the improvements of the quality and speed of the Internet have transformed our society into one that depends strongly on the quick access to information. The huge amount of data that is generated by this process of communication is accumulated daily and is stored in form of text documents, databases etc. The retrieving of this data is not simple and therefore the data mining techniques were developed for extracting information and knowledge represented in patterns or concepts that are sometimes not obvious. Therefore automated document classification is an important challenge. In order to facilitate document retrieval and analysis it is essential to be able to automatically organize documents into classes. One general procedure for this classification is to take a set of pre-classified documents and consider them as the training set. The training set is then analyzed in order to obtain a classification scheme. Such a classification scheme often needs to be refined with a testing process. After that, this scheme can be used for classification of other on-line documents. The classification analysis decides which attribute-value pairs set has the greatest discriminating power in determining the classes [4]. Text classification is a general process that includes a lot of requirements that need to be fulfilled in order to solve the problem. Some of those requirements have a high influence on the final accuracy of classification. In the last years a lot of research efforts are centered on automatically document classification.

The process of knowledge discovery in database, in our content called text mining, has more steps: preprocessing (attributes extraction, attributes selection), effectively data mining, pattern evaluation and knowledge presentation. In this paper we focus on the attributes selection step only.

Preprocessing step is important because, in this step, the data are prepared for better knowledge extraction and therefore the quality of learning process is improved.

Analysis and mining of large amounts of data can take a long time or may be impossible due to the size of the data. In this context data reduction techniques are applied in order to obtain a new dataset, with a reduced representation of the original dataset but with a distribution close to the original data. In case of using the reduced dataset, the data mining process can be more efficient and may produce the same (or almost the same) analytical results, into a shorter time. In the text mining process attribute selection is named feature selection and a lot of methods for selecting the best attributes are proposed.

1.1 Analyzing Text Data and Information Retrieval

Information retrieval (IR) is concerned with the organization and retrieval of information from a large number of text-based documents. A typical information retrieval problem is to locate relevant documents based on user input, such as keywords or example documents. Usually information retrieval systems include on-line library catalog systems and on-line document management systems. Since information retrieval and database systems each handles different kinds of data, there are some database system problems that are usually not present in information retrieval systems: concurrency control, recovery, transaction and management. There are also some common information retrieval problems that are usually not encountered in traditional database systems: unstructured documents, approximate search based on keywords and the notion of relevance.

1.2 Basic Measures for Text Retrieval

May [Relevant] be the set of documents relevant to a query and [Retrieved] be the set of documents retrieved. The set of documents that are both relevant and retrieved is denoted by $[\text{Relevant}] \cap [\text{Retrieved}]$. There are two basic measures for assessing the quality of text retrieval:

- *Precision*: is the percentage of retrieved documents that are in fact relevant to a query. It is defined as follows:

$$precision = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|} \tag{1}$$

- *Recall*: is the percentage of documents that are relevant to the query and were in fact retrieved.

$$recall = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|} \tag{2}$$

- *F-measure* is defined as harmonic mean between Precision and Recall

$$F - measure = \frac{recall \times precision}{(recall + precision)/2} \tag{3}$$

- *Accuracy:* is the percent of documents correctly classified in the classes based on the document target (label).

Precision ranges from 1 (all retrieved documents are relevant) to 0 (none of relevant document is retrieved). *Recall* range from 1 (all relevant documents are retrieved) to 0 (none of retrieved document is relevant). In fact *precision* represents a quantitative measure of the information retrieval system while *recall* represents a qualitative measure of this system.

1.3 Keyword-Based and Similarity-Based Retrieval

The information retrieval system based on similarity measures finds similar documents based on a set of common keywords. The output for this system is based on the degree of relevance measured based on using keywords closeness and the relative frequency of the keywords. In some cases it is difficult to give a precise measure of the relevance between keyword sets. In modern information retrieval systems keywords for document representation are automatically extracted from the document. This system often associates a stoplist with the set of documents. A stoplist is a set of words that are deemed "irrelevant" and can vary when the document set varies. Another problem that appears is *stemming*. A group of different words may share the same word stem. A text retrieval system needs to identify groups of words where the words in a group are small syntactic variants of one another, and collect only the common word stem per group.

Let's consider a set of *d* documents and a set of *t* terms for modeling information retrieval [2]. We can represent each document as a vector v in the t dimensional space $\mathbb{R}^t$. The i^{th} coordinate of v (noted v_i) is a number that measures the association of the i^{th} term with respect to the given document: it is generally defined as 0 if the document does not contain the term, and nonzero otherwise. The $\underline{v_i}$ indicates the frequency of the term in the document and there are a lot of methods to define term frequency. Similar documents are expected to have similar relative term frequency, and we can measure the similarity among a set of documents or between a document and a query.

1.4 Unsupervised versus supervised learning

As mentioned in [5, 8], machine learning provides the basic techniques for data mining by extracting information from raw data contained in databases. Machine learning techniques are divided into two sub domains: unsupervised learning and supervised learning.

In unsupervised learning the algorithm receives only data without the class label (called unlabeled data) and the algorithm task is to find an adequate representation of data distribution.

In supervised learning, the algorithm receives data (the text documents) and the class label for the corresponding classes of the documents (called labeled data). The purpose of supervised learning is to learn the concepts that correctly classify documents for given classification algorithm. Based on this learning the classifier will be able to predict the correct class for unseen examples. Under this approach, the over-fitting effects may appear. This will happen when the algorithm memorizes all the labels for each case.

The accuracy of supervised learning is usually assessed on a test set of examples disjoint from the training set examples. Classification methods used varies, ranging from traditional statistical approaches, neural networks to kernel type algorithms [7].

Some researchers have combined unsupervised and supervised learning that has emerged the concept of semi-supervised learning [6]. In this approach an unlabeled dataset is initially analyzed

in order to make some assumptions about data distribution and later this hypothesis is confirmed or rejected by a supervised approach.

1.5 Data transformation - Normalization

After representing the data as vector of attributes, the attributes are in different value domains and need to be transformed in the same domain. Therefore we have to apply some normalization methods for scaling the values into a specific domain. For example the values are scaled usually into [-1.0, 1.0] or [0.0, 1.0] domain.

- Binary representation – in the vector we store "0" if the word doesn't occur in the document and "1" if it occurs without considering the number of occurrences. The weight can only be 0 or 1.
- Nominal representation – in the vector we compute the value of the weight using the formula:

$$TF(d,t)=\frac{n(d,t)}{\max_{\tau} n(d,\tau)} \tag{4}$$

  where n(d, t) is the number of times that term t occurs in document d, and the denominator represents the value of term that mostly occurs in document d, and $TF(d,t)$ is the term frequency. The weight can take values between 0 and 1.

- Min-max representation – Consider that min_A and max_A are the minimum and the maximum values for attribute A. Transforming the value v of A in v' that will be in new domain $[new\_min_A, new\_max_A]$ can be make:

$$v`=\frac{v-min_A}{max_A\text{-}min_A}(new\_max_A\text{-}new\_min_A)+new\_min_A \tag{5}$$

- Z-mean normalization – the value of attribute A is transformed based on mean and the standard deviation thus:

$$v`=\frac{v-\overline{A}}{\sigma_A} \tag{6}$$

  where ν is the value of A, ν' is the new value, $\overline{A}$ represent the mean and σ_A is standard deviation. This method is recommended when the min and the max of attribute is unknown.

From normalization methods we have used the Nominal representation.

2 Feature selection methods

In this research we have tested tree different feature selection methods.

2.1 OneR Attribute Selection

In [7] we have presented some features selection methods based on advanced learning algorithms. A problem of these was the long training time and, after that, the selection of relevant features. Using methods based on learning algorithms usually the quality of selected features is better and therefore the learning quality is improved. Based on this idea we have experimented and present in

this article a method based on a simple and fast classifier algorithm. This method evaluates each attribute individually by using the 1R classifier. The rule of this classifier is based only on the attribute values and the topic. Thus, for each attribute and for each value of the attribute, the error produced if only that attribute will be used to classify the dataset is computed. After that, the attribute with the smallest number of errors is chosen. As a feature selection method the algorithm will sort descending the attributes based on the error rate obtained (by each attribute independently) and retains the first desired number of attributes.

The pseudo code for the 1R classifier is presented below:

```
for each attribute
{  for each value of that attribute
   { compute the class distribution based on attribute value
     Class_label = select most frequent class
     create a rule: attribute = value => Class_label
   }
   Calculate the error rate of the rule on the whole dataset
}
Select rule with lowest error rate
```

2.2 Information Gain

Information Gain and Entropy [6] are functions of the probability distribution that underlie the communication theory. The entropy is a measure of uncertainty of a random variable. Based on entropy, for features selection a measure called "Information Gain" is defined. This represents the expected reduction in Entropy caused by partitioning the samples according to this attribute. The Information Gain of an attribute relative to a collection of samples S, is defined as:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{7}$$

where *Values(A)* is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A is equal to v.

Forman in [2] reported that Information Gain failed to produce good results on an industrial text classification problem, as Reuter's database. The author attributed this to the property of many feature scoring methods to ignore or to remove features needed to discriminate difficult classes. In this paper we intent to revalidate this in our dataset context.

2.3 Gain Ratio

The Gain Ratio method, called also Information Gain Ratio, is just the ratio between the information Gain and the intrinsic features value. This wants to correct the problem of Information Gain method when we have an attribute with high value comparatively with others attributes values.

$$GR(S, A) = \frac{Gain(S, A)}{-\sum_{v \in Values(A)} \frac{|S_v|}{|S|} * \log_2 \frac{|S_v|}{|S|}} \tag{8}$$

In our experiments all features are normalized using nominal representation and this big difference occurs only in case when a word appears in a large numbers of times in the same document comparatively with others words. Given that we are working with documents, and this possibility occurs very rarely, we expected this feature selection method to return worse results comparatively

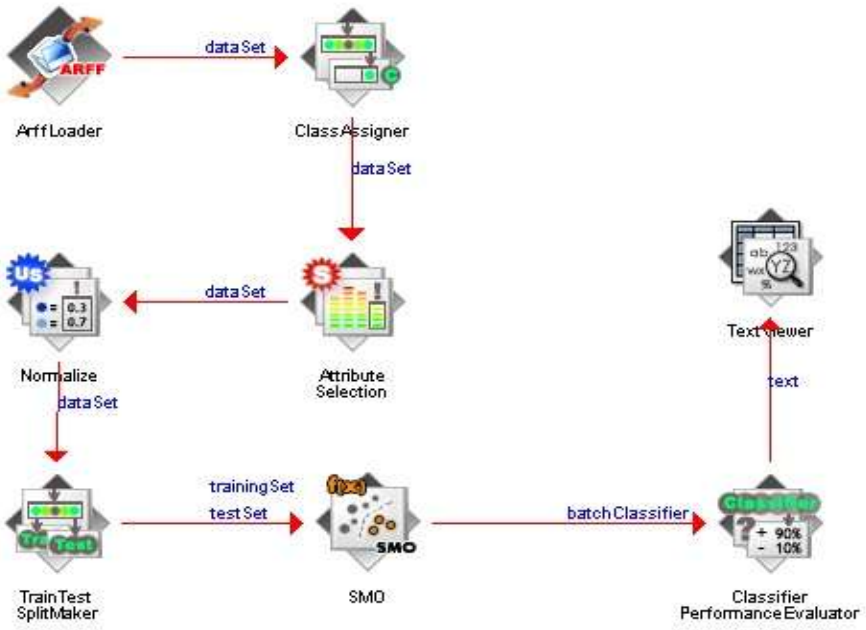

Figure 1. The WEKA project

with information gain. Therefore the gain ratio score divides the score of information gain with intrinsic features value:

3 Experimental Framework

3.1 The Dataset

Our experiments were performed on the Reuters-2000 collection [9], which has 984 Mb of newspapers articles in a compressed format. The collection includes a total of 806,791 documents, with news stories published by Reuters Press covering the period from 20.07.1996 through 19.07.1997. Documents are pre-classified according to 3 categories: by the Region the article refers to, by Industry Codes and by Topics proposed by Reuters (126 topics, 23 of them contain no articles). Due to the huge dimensionality of the database we will present here results obtained using a subset of data. From all documents we have selected the documents for which the industry code value is equal to "System software". We obtained 7083 documents that are represented using 19038 features and 68 topics. After applying a stop-word filter (from a standard set of 510 stop-words) and extracting the word stem [1, 3] we have represented a document as a vector of words. From these 68 topics we have eliminated those topics that are poorly or excessively represented. Thus we have eliminated those topics that contain less than 1% documents from all 7083 documents in the entire set. We have also eliminated topics that contain more than 99% samples from the entire set, as being excessively represented. Also for each document we have considered only the first topic, as label by Reuters. After doing so we have obtained 16 different topics and 7053 documents, that were split randomly in a training set (4702 samples) and a testing set (2351 samples). In the feature extraction phase we take into consideration both the article and the title of the article. In the feature selection phase we have selected a different number of features, from 200 to 7000.

No. features	Information Gain	Gain Ratio	OneR	No. features	Information Gain	Gain Ratio	OneR
200	92.45	88.82	90.03	**1700**	93.33	91.45	93.37
300	92.87	88.95	91.62	**1800**	93.54	91.74	93.41
400	93.37	89.45	91.87	**1900**	93.58	91.99	93.04
500	92.99	89.57	91.99	**2000**	93.29	91.91	93.08
600	93.12	89.49	92.41	**2500**	92.62	92.29	92.70
700	92.74	89.62	92.08	**3000**	92.70	92.20	92.66
800	93.24	89.53	92.08	**3500**	92.95	93.04	92.41
900	93.45	89.45	92.54	**4000**	92.37	92.37	92.91
1000	93.16	89.99	92.70	**4500**	92.41	92.41	92.62
1100	93.49	89.99	92.95	**5000**	92.54	92.54	92.45
1200	93.45	90.08	92.99	**5500**	92.70	92.70	92.54
1300	93.24	90.91	92.79	**6000**	92.74	92.74	92.62
1400	93.08	91.03	93.20	**6500**	92.62	92.62	92.95
1500	93.45	91.03	92.99	**7000**	92.91	92.91	92.91
1600	93.29	91.12	93.16				

Table 1. Experimental Results (accuracy)

3.2 Weka framework

For feature selection and classification steps we use WEKA KnowledgeFlow Enviroment [10]. The project flowchart realized in weka is presented in Figure 1. In the *AttributeSelection* component we have chosen *OneRAttributeEval*, *InfoGainAttributeEval* and *GainRatioAttributeEval*. In the *Normalize* component we have used the nominal representation. In the *TrainTestSplitMaker* we have chosen 66% *trainPercent*. As classifier algorithm (in *SMO* component) we use the Sequential Minimal Optimization (SMO) implementation of Support Vector Machine (SVM) method that we have also presented in [7]. As kernel we use the *PolyKernel* with exponent c=1 (linear kernel).

4 Experimental Results

In text classification problems usually we have a great number of features that are extracted from the dataset in order to create the input vector. Usually many of those features are rather irrelevant in classification. These features don't generally improve the accuracy of the classification and only increase the training and testing time and the memory requirement. For a fair comparison between the three feature selections methods used, we need to use the same number of features. For all methods we have a threshold that represents the number of features that we want to be obtained. In Table 1 and Figure 2 we present accuracy results obtained for each features selection methods and for a number of features between 200 and 7000 obtained from the Reuters Database [9]. We have evaluated also Precision and Recall measures.

OneR learning algorithm obtains better results than GainRatio for a small number of features. When the features number increases the results are approximately the same with the other algorithms. The difference between the minimum accuracy value (90.03%) and the maximum value (93.41%) is relatively small. The maximum value was obtained for 1800 features and after 3500 features the accuracy has even a small decrease. This demonstrates that with a good features selection method

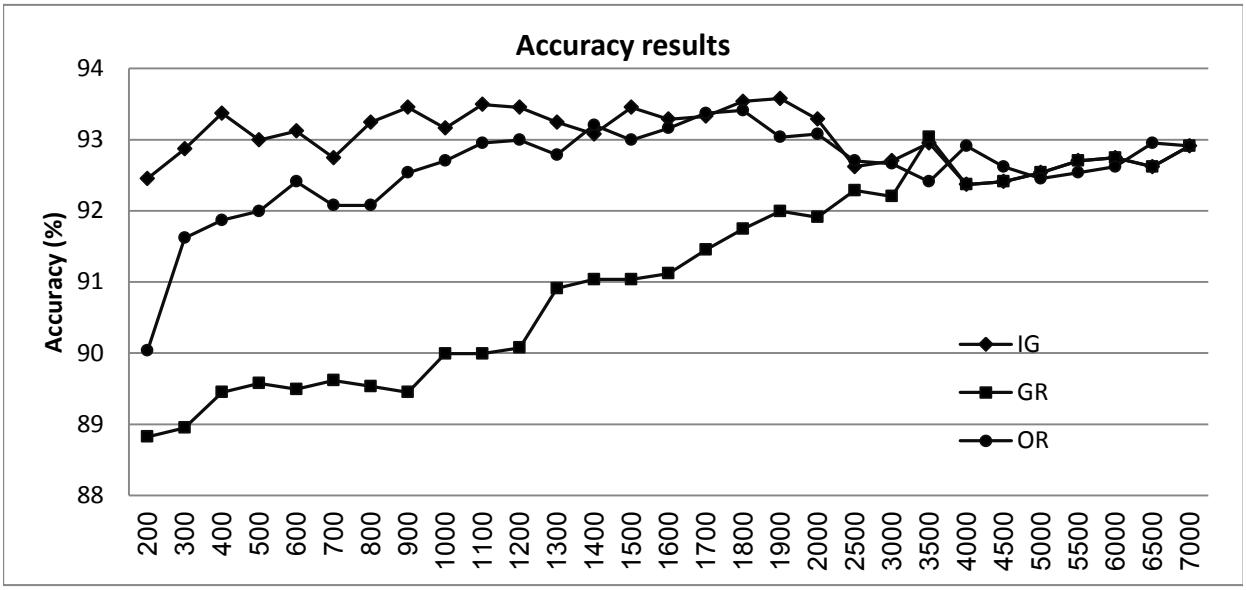

Figure 2 Classification accuracy obtained for different number of features

we can obtain good result for a small number of features and also with a short learning time and reduced memory usage.

For Gain Ratio method the classification accuracy had a greater increase from 88.82% to 93.04%.

Also the precision metric for this method increases from 0.898 to 0.93 and the recall metric had approximately the same increasing from 0.888 to 0.93. These indicate that the classifier has no significant difference when we change between the training and testing set. For a small number of features this method does not select the best attributes, but when the number of attributes increases this method succeed to reach comparatively results with the other methods. When the number of feature increases to more than 3500 the classification accuracy has not increased, sometimes even it decrease. The best accuracy result was obtained for 3500 features. As we have expected, because this method divide the IG with an intrinsic feature values, this method obtains the worst results in our tests. This happen because we don't have very unbalanced attributes.

The best results were obtained with the information gain method. We say the best results because the difference between the minimum accuracy (92.37%) and the maximum accuracy (93.5%) is small and because values, close to maximum value, were obtained with a small number of features. For only 900 features this method already obtains an accuracy value of 93.46% that is close to the maximum obtained value. When the number of features increases to more than 1900 features, the accuracy starts to decrease. With this method we obtain the best accuracy value for a number of 1900 features (that represents 27.14% of all features). When the numbers of features increase above 3500 all the presented methods obtain approximately the same results.

5 Conclusions

In this paper we have evaluated three features selection methods in the text document classification problem. Our experiments where done based on the WEKA package. As classifier algorithm we use the SMO implementation of Support Vector Machine algorithm, based only on a polynomial kernel with degree 1, which means a linear kernel. We use only this type of kernel because we consider

that in general the text document classification problem is a linear problem. Using Gain Ratio, a method that obtains better results comparatively with IG in some other classification problems, in the text document classification problem we found that this advantage does not appear, even it leads to worse results.

Using a feature selection method based on the 1R classifier the accuracy increases but the best values were obtained using a classical implementation of the information gain IG method. Also for the OneR and IG feature selection methods the best results were obtained for a small number of features (1800 features for OneR method and 1900 features for IG method). The best accuracy result was obtained with the IG method for 1900 features and it was 93.58%.

Our next challenge is to classify larger text data sets (the complete Reuters database). We try to develop a pre-classification of all documents, obtaining fewer samples (using simple algorithms like Linear Vector Quantization or Self Organizing Maps). After that we'll use only the obtained samples as input vectors for the already implemented features selection and classification methods.

6 Bibliography

[1] Bhatia, S., *Selection of Search Terms Based on User Profile*, ACM Explorations, pages. 224-233, 1998.

[2] G. Forman, "A Pitfall and Solution in Multi-Class Feature Selection for Text Classification", Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, 2004.

[3] Mladenic, D., *Feature Subset Selection in Text Learning*, Proceedings of the 10th European Conference on Machine Leraning (ECML-98), pages 95-100, 1998.

[4] Mladenic, D., Grobelnik, M. Feature selection for unbalanced class distribution and naïve bayes, In Proceedings of the 16th International Conference on Machine Learning ICML, p.258-267,1999.

[5] Mladenic, D., Brank, J., Grobelnik, M., Milic-Frayling, N., Feature Selection Using Support Vector Machines The 27th Annual International ACM SIGIR Conference (SIGIR2004), pp 234-241, 2004.

[6] T. Mitchell, "Machine Learning", McGraw Hill Publishers, 1997.

[7] D. Morariu, Text Mining Methods based on Support Vector Machine, MatrixRom, Bucharest, 2008.

[8] J. Platt, "Fast training of support vector machines using sequential minimal optimization". In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods – Support Vector Learning, pages 185-208, Cambridge, MA, 1999, MIT Press.

[9] Reuters Corpus: http://about.reuters.com/researchandstandards/corpus/. Released in November 2000

[10] WEKA package - http://www.cs.waikato.ac.nz/ml/weka/index.html (tacked in 2010)