Building a Topic Hierarchy Using the Bag-of-Related-Words Representation

Rafael Geraldeli Rossi
Institute of Mathematics and Computer Science - University of São Paulo
P.O. Box 668
São Carlos, SP - Brazil
ragero@icmc.usp.br

Solange Oliveira Rezende
Institute of Mathematics and Computer Science - University of São Paulo
P.O. Box 668
São Carlos, SP - Brazil
solange@icmc.usp.br

ABSTRACT

A simple and intuitive way to organize a huge document collection is by a topic hierarchy. Generally two steps are carried out to build a topic hierarchy automatically: 1) hierarchical document clustering and 2) cluster labeling. For both steps, a good textual document representation is essential. The bag-of-words is the common way to represent text collections. In this representation, each document is represented by a vector where each word in the document collection represents a dimension (feature). This approach has well known problems as the high dimensionality and sparsity of data. Besides, most of the concepts are composed by more than one word, as "document engineering" or "text mining". In this paper an approach called bag-of-related-words is proposed to generate features compounded by a set of related words with a dimensionality smaller than the bag-of-words. The features are extracted from each textual document of a collection using association rules. Different ways to map the document into transactions in order to allow the extraction of association rules and interest measures to prune the number of features are analyzed. To evaluate how much the proposed approach can aid the topic hierarchy building, we carried out an objective evaluation for the clustering structure, and a subjective evaluation for topic hierarchies. All the results were compared with the bag-of-words. The obtained results demonstrated that the proposed representation is better than the bag-of-words for the topic hierarchy building.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithm, Experimentation

DocEng'11, September 19-22, 2011, Mountain View, California, USA.
Copyright 2011 ACM 978-1-4503-0863-2/11/09 ...$10.00.

Keywords

Document Representation, Text Mining, Topic Hierarchy

1. INTRODUCTION

Part of the data in the digital universe is in the textual format, like e-mails, reports, papers, and web-pages contents. To manage, search and browse in large text collections, topic hierarchies are very useful. Topic hierarchies are structural organizations where the documents are separated in a tree in which the nodes close to the root are general topics and the nodes far from the root are more specific topics. This type of structure allows an exploratory search. The user can browse in the textual document collection interactively through the topics of the nodes. Examples of topic hierarchies are the online directories of Yahoo[1] and Open Directory Project[2].

The manual building of a topic hierarchy for large text collections requires a huge human effort. Therefore methods to perform this task automatically have received great attention in the literature [13, 15, 9]. Generally hierarchical clustering methods are used to create the hierarchical structure organization automatically, and then a process of cluster labeling is performed. For both steps, the textual document collection has to be represented in an appropriate way.

The Vector Space Model (VSM) [14] is generally used to represent textual documents. In this model, each document is represented by a vector, and each position of this vector is a dimension (feature) of the document collection. A common approach based on the VSM is the bag-of-words. In this representation, each word in a textual document collection becomes a dimension in the vector space. However, this approach presents some problems such as the high dimensionality, the high sparsity data, and the word order is not kept.

Another problem presented by the bag-of-words is that concepts are usually represented by more than one word, such as "document engineering", "topic hierarchy", and "text mining". For instance, if we look at the features "machine learning" or "data mining", there is no doubt about the topic of the document. On the other hand, features compounded by single words as "learning" can represent documents about machine learning or about teaching, and "mining" can represent documents about data mining or mineral extraction.

Trying to obtain features compounded by more than one

[1] http://dir.yahoo.com/
[2] http://www.dmoz.org/

word, approaches were developed based on n-grams [5, 17, 4, 12], and based on set of words [10, 22, 19, 6, 21]. These approaches usually add the features to the bag-of-words, increasing the already high dimensionality. Some approaches analyze the entire collection to generate the features. This can have a high computational cost due the dimensionality, and generate features without meaning. Furthermore, most of the approaches perform a supervised feature selection, requiring labeled collections, which is not common in real world textual document collections.

This paper proposes an approach that generates related words and use them as features. The proposed approach, named *bag-of-related-words*, generates features from each document of a collection through association rules. The association rules [1] are used to discover relations among items in a dataset. Besides obtaining the relations of the words (items), the intention of using association rules is to reduce the dimensionality of the bag-of-words. This is possible because not all the words of a document will be used as features, but only the words that occur or cooccur with a frequency above a certain threshold.

In order to extract association rules from each document, a mapping of the document into transactions is necessary. This can be done considering the sentences, paragraphs, or a sliding window as transactions. Since the words are related in specific contexts of the documents, the mapping can produce more understandable features to the user. An analysis will be carried out on which way the mapping of the document into transactions produces better results.

Interest measures can be applied to prune the association rules. In this paper the Confidence, Lift, Yules'Q, Linear Correlation Coefficient, Mutual Information, Gini Index, J-Measure, and Kappa measures [8, 7, 3, 18] were used. An evaluation about which measure produces better results for the textual document clustering was carried out.

A representation is created based on the VSM with the features generated for each textual document. An hierarchical clustering is built and a process of cluster labeling is performed. Experiments were carried out to analyze the feasibility of the bag-of-related-words to build a hierarchical cluster and its contribution to the cluster labeling. When the bag-of-related-words were used to represent the documents, the clustering quality and the topics of the hierarchy were better than the bag-of-words.

This paper is organized as follows. Section 2 presents the related works about textual document representation using features compounded by more than one word. Section 3 presents the details of the proposed approach. Section 4 shows the results of the proposed approach and the bag-of-words for the textual document clustering and topic taxonomy labeling. Finally, Section 5 presents the conclusions and future works.

2. TEXTUAL DOCUMENT REPRESENTATION

The textual document representation is fundamental to build a topic hierarchy automatically or for some other pattern extraction from textual documents. Besides the bag-of-words representation, there are several works that generate features compounded by more than one word. Usually these features can be generated using n-grams (or statistical phrases) or using set of words.

N-grams are sequences of n words that appear in the text collection. Some works using n-grams are [5, 17, 4, 12]. A general characteristic of these approaches is that the use of all the sequences of n words produces a dimension much higher than the bag-of-words. Some approaches try to reduce the number of generated word sequences. For instance, in [17] the bigrams are generated if at least one of the components is frequent in a document.

Another way to represent textual documents is through features compounded by set of words. The words that compound this type of feature do not necessarily occur together or close to each other in a textual document. This representation can be considered an extension of the n-grams. Some works using set of words as features are [10, 22, 19, 6, 21]. Most of the works generally execute an cooccurrence analysis in a bag-of-words to generate features compounded by sets of words. Some approaches try to avoid this type of analysis. For instance, in the proposed approach by Lie. et al. [22], the "loose n-grams" were defined. The loose n-grams are set of words that cooccur in a limited space, as sentences, or a sequence of words (window). The relative position of the words are not considered. Only the words that appear in a minimum number of documents are used. The words that have a good discriminative power are separated. Then, sets of words compounded by words that cooccur in the collection are obtained, and these sets of words are filtered using the χ^2 method. The results demonstrated a little improvement in relation to the bag-of-words.

Most of the related researches about features compounded by more than one word generate a number of features much higher than the bag-of-words and most of the time a process of feature selection is necessary. This is computationally expensive due the high dimensionality of the collection and the need of labeled collections, which is not common in real textual document collections.

Moreover, some approaches that analyze the entire collection may obtain features without meaning. For instance, if we consider a collection of scientific articles, the feature "*introduction_conclusion*" can be generated. Besides, collections with few documents of some vocabulary may not have their features generated. For instance, considering 5000 documents about chemistry and 20 about computer science. Supposing that some words of the documents about computer science cooccur only in 5 documents, the minimum cooccurrence threshold must be of 0.001. On the other hand, a huge number of features about chemistry could be generated.

3. BAG-OF-RELATED-WORDS

The proposed approach, called bag-of-related-words, generates features compounded by a set of related words from textual documents. The main goal of this approach is to use related words that repeat over the document in limited spaces as features.

The related words are obtained through association rules. An association rule is a rule of the type $A \Rightarrow B$, in which A and B are groups of items, called itemsets, and $A \cap B = \emptyset$ [1]. The association rules discover relations among the itemsets in a dataset, in which $A \Rightarrow B$ means that when A occurs, B also tends to occur. Two classical measures to generate association rules are Support and Confidence. Support measures the joint probability of an itemset in a database, that is, $sup(A \Rightarrow B) = q(A \cup B)/Q$, in which $q(A \cup B)$ is the number of transactions that A and B occur together, and Q is the total number of transactions. Confidence indicates the probability of A and B occur together given that A occurred, that is, $conf(A \Rightarrow B) = q(A \cup B)/q(A)$. Usually minimum values of support and confidence are defined to generate the rules.

The four main steps to generate the features from the textual documents with the proposed approach are:

1. Mapping the textual document into transactions;
2. Extracting association rules from the transactions;
3. Using the itemsets of the rules to compound the features;
4. Using the features to construct the *document-term* matrix;

The mapping of the textual document into transactions allows the extraction of association rules. In this step, Text Mining preprocessing tasks as stopwords removal, and stemming can be applied. After the preprocessing, the transactions can be obtained considering the sentences, paragraphs, or a sliding window.

When sliding windows are used to map the transactions, the first transaction contains only the first word, the second transaction contains the two first words, and so on, until the window contains the number of words equal the defined size (l). After this, the sliding window slides one word and considers the next l words of the document. At the end of the document, the last (l) transaction contains the last l words, the last $l - 1$ transaction contains the $l - 1$ words, and so on, until the window contains only one word. This is done so that all the words are contained the same number of times in a sliding window.

The second step consists of extracting the association rules from the transactions which were mapped from the text. To illustrate the two first steps, consider the content of the Table 1. There is a text about Data Mining taken from Wikipedia[3], that we called "example text". Applying the stopword removal and stemming, and considering the sentences as transactions, it was possible to map the text into 10 transactions. Considering these 10 transactions, a minimum support threshold of 30, 0% and a minimum confidence threshold of 75%, 15 association rules were extracted. The values on the right of the rules are the support and the confidence respectively. It can be noticed that the itemsets of the rules on Table 1 are really related to the example text, as *"data"* and *"mine"*, *"larger data"* and *"sampl"*.

The proposed approach also considers features compounded by single words. The items of the rules with empty set like $\emptyset \Rightarrow data$ will be used as features compounded by single words. In this case the feature "*data*" would be generated.

The different ways to map the document into transactions can modify the number of transactions, the frequency of the words and their cooccurrences. The frequency of the words is modified due the number of repetitions of the same word in the transaction. For instance, the word *"data"* occurs 15 times in the example text from Table 1. If we consider the paragraphs as transactions, the word *"data"* occurs 11 times in the first transaction, and is counted only once. Also, the mapping of paragraphs made the word *"data"* occurs in all the transactions (2 transactions), that is, the word *"data"* has a support of 100%. If we consider the sentences as transactions, the word *"data"* occurs at most three times in a transaction and occurs in 7 of 10 transactions. Thus its frequency is equal to 7 and its support is equal to 70%.

The cooccurrences of the words are also modified by the type of transaction. For instance, the mapping considering the paragraphs make the words cooccur more with other words than the mapping of sentences. Different values of minimum support are required due all the modifications caused by the different types of mapping. To illustrate the practical impact of the mapping, we present in Table 2 the extracted frequent itemsets considering the example text from Table 1. We used the following types of transactions and their respective minimum supports: sentences 30.0%, paragraphs 100.0%, size 5 sliding window 10.0%, size 10 sliding window 21.0%, size 20 sliding window 47.0%, and size 30 sliding window 58.0%. These minimum support values were chosen to generate a similar number of frequent itemsets.

It can be noticed that some itemsets, that are useful to identify the example text from Table 1, appear in all the types of mapping as *"process"*, *"mine"*, *"pattern"*, *"data"*, *"mine_pattern"*, *"data_mine"*, *"data_pattern"*, and *"data_mine_pattern"* (highlighted with italic font). Some other useful itemsets do not appear in all the types of mapping as *"mine_sampl"* (only in sentences, sliding windows of size 5, 10, and 20), *"pattern_sampl"* (it does not appear only in paragraphs), *"data_mine_sampl"* (only in sentences, and sliding windows of size 10 and 20), *"data_mine_process"* (only in paragraphs and size 30 sliding window), *"mine_pattern_process"* (only in paragraphs and size 30 sliding window), and *"data_mine_pattern_process"* (only in the size 30 sliding window). Given the differences of each type of mapping, this paper will analyze which mapping produces better results for the textual document clustering.

The third step consists of using the itemsets of the association rules obtained in the previous step to compound the features of a document. For instance, the rule "*data ⇒ mine*" will generate the feature "*data_mine*". In order to avoid that rules with the same itemsets generate different features, the items of the rule are sorted lexicographically or according to the order they occur in the textual document.

In this step, interest measures besides support and confidence can be used to obtain different relations among the itemsets, to rank and prune the association rules. The intention to use interest measures is to obtain more understandable features and reduce even more the dimensionality. The Confidence, Lift, Yules'Q, Linear Correlation Coefficient, Mutual Information, Gini Index, Kappa, and J-Measure measures [8, 7, 3, 18] were used.

The ranking and consequently the pruning of the rules using these measures can be different due their characteristic. To illustrate this, on Table 3 are presented the rankings of features considering a text about clustering validation measures [20] with the Confidence, Mutual Information and Kappa measures. Important features to describe the textual document appear in the rankings of these 3 measures as "*data set*", "*measur valid*", and "*cluster valid*". Other important features appear only in the rankings of the Confidence and Mutual Information measures as "*contig matrix*", "*data sample*", and "*extern valid*". Other features appear only in the ranking of one measure, as "*cluster data*" that

[3] http://en.wikipedia.org/wiki/Data_mining (May 13, 2010).

Table 1: Process of association rule extraction from a textual document.

Example Text
Data mining is the process of extracting patterns from data. Data mining is becoming an increasingly important tool to transform this data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery. Data mining can be used to uncover patterns in data but is often carried out only on samples of data. The mining process will be ineffective if the samples are not a good representation of the larger body of data. Data mining cannot discover patterns that may be present in the larger body of data if those patterns are not present in the sample being ”mined”. Inability to find patterns may become a cause for some disputes between customers and service providers. Therefore data mining is not foolproof but may be useful if sufficiently representative data samples are collected. The discovery of a particular pattern in a particular set of data does not necessarily mean that a pattern is found elsewhere in the larger data from which that sample was drawn. An important part of the process is the verification and validation of patterns on other samples of data.

Transactions considering the sentences of the preprocessed example text.
1. data mine process extract pattern data 2. data mine increasingli import tool transform data inform 3. commonli wide rang profil practic market surveil fraud detect scientif discoveri 4. data mine uncov pattern data carri sampl data 5. mine process ineffect sampl good represent larger bodi data 6. data mine discov pattern present larger bodi data pattern present sampl mine 7. inabl find pattern disput custom servic provid 8. data mine foolproof suffici repres data sampl collect 9. discoveri pattern set data necessarili pattern found larger data sampl drawn 10. import part process verif valid pattern sampl data

Rules obtained considering the sentences as transactions.		
data ⇒ ∅ (80.0, 80.0)	data ⇒ mine (60.0, 75.0)	mine pattern ⇒ data (30.0, 100.0)
process ⇒ data (30.0, 100.0)	data ⇒ sampl (60.0, 75.0)	larger data ⇒ sampl (30.0, 100.0)
larger ⇒ sampl (30.0, 100.0)	sampl ⇒ data (60.0, 100.0)	pattern data ⇒ sampl (40.0, 80.0)
larger ⇒ data (30.0, 100.0)	mine ⇒ data (60.0, 100.0)	mine sampl ⇒ data (40.0, 100.0)
pattern ⇒ data (50.0, 83.3)	larger sampl ⇒ data (30.0, 100.0)	pattern sampl ⇒ data (40.0, 100.0)

Table 2: Frequent itemsets obtained through different ways of mapping the text from Table 1 into transactions.

Sentence

Itemsets	**Sup.**
larger	30,0
process	30,0
mine	60,0
pattern	60,0
sampl	60,0
data	80,0
data process	30,0
data larger	30,0
larger sampl	30,0
mine pattern	30,0
mine sampl	40,0
pattern sampl	40,0
data pattern	50,0
data mine	60,0
data sampl	60,0
data larger sampl	30,0
data mine pattern	30,0
data mine sampl	40,0
data pattern sampl	40,0
-	-

Paragraph

Itemsets	**Sup.**
data	100,0
mine	100,0
pattern	100,0
process	100,0
data mine	100,0
data pattern	100,0
data process	100,0
mine pattern	100,0
mine process	100,0
pattern process	100,0
data mine pattern	100,0
data mine process	100,0
data pattern process	100,0
mine pattern process	100,0
data mine pattern process	100,0
-	-
-	-
-	-
-	-
-	-

Size 5 Window

Itemsets	**Sup.**
bodi	10,9
discoveri	10,9
import	10,9
present	10,9
larger	16,3
process	16,3
sampl	32,6
mine	38,0
pattern	42,4
data	69,6
mine sampl	9,8
pattern present	9,8
data larger	12,0
pattern sampl	14,1
mine pattern	15,2
data sampl	21,7
data mine	30,4
data pattern	30,4
data pattern sampl	9,8
data mine pattern	10,9

Size 10 Window

Itemsets	**Sup.**
larger	28,9
process	30,9
sampl	54,6
mine	63,9
pattern	67,0
d*ata*	94,8
larger pattern	21,6
larger sampl	21,6
pattern process	21,6
data larger	28,9
data process	30,9
mine sampl	30,9
pattern sampl	42,3
mine pattern	44,3
data sampl	52,6
data mine	61,9
data pattern	64,9
data larger pattern	21,6
data larger sampl	21,6
data pattern process	21,6
data mine sampl	28,9
data pattern sampl	40,2
data mine pattern	42,3

Size 20 Window

Itemsets	**Sup.**
process	56,1
sampl	70,1
mine	75,7
pattern	90,7
data	100,0
mine sampl	47,7
pattern process	54,2
data process	56,1
pattern sampl	69,2
mine pattern	69,2
data sampl	70,1
data mine	75,7
data pattern	90,7
data mine sampl	47,7
mine pattern sampl	47,7
data pattern process	54,2
data pattern sampl	69,2
data mine pattern	69,2
data mine pattern sampl	47,7
-	-
-	-
-	-
-	-

Size 30 Window

Itemsets	**Sup.**
sampl	72,6
process	76,9
mine	77,8
pattern	94,9
data	100,0
mine process	59,8
pattern sampl	71,8
data sampl	72,6
mine pattern	75,2
pattern process	75,2
data process	76,9
data mine	77,8
data pattern	94,9
mine pattern process	58,1
data mine process	59,8
data pattern sampl	71,8
data mine pattern	75,2
data pattern process	75,2
data mine pattern process	58,1
-	-
-	-
-	-
-	-

Table 3: Ranking of the features for the Confidence, Mutual Information, and Kappa measures.

Confidence			Mutual Information			Kappa		
Rank	**Features**	**Score**	**Rank**	**Features**	**Score**	**Rank**	**Features**	**Score**
1º	conting matrix	0,766	1º	data set	0,148	1º	measur valid	0,427
2º	effect uniform	0,663	2º	conting matrix	0,100	2º	measur normal	0,384
3º	effect mean uniform	0,600	3º	effect uniform	0,083	3º	cluster measur	0,361
4º	bound upper	0,562	4º	bound upper	0,073	4º	measur properti	0,351
5º	data set	0,552	5º	inform mutual	0,055	5º	extern measur	0,339
6º	cluster measur valid	0,336	6º	data sampl	0,049	6º	entropi measur	0,333
7º	inform mutual	0,331	7º	measur valid	0,047	7º	evalu measur	0,330
8º	rand statist	0,271	8º	rand statist	0,039	8º	mean measur	0,329
9º	measur valid	0,237	9º	data imbalanc	0,033	9º	measur select	0,328
10º	data sampl set	0,226	10º	extern valid	0,032	10º	cluster valid	0,328
11º	cluster valid	0,207	11º	data sampl set	0,031	11º	consist measur	0,327
12º	cluster mean measur	0,203	12º	effect mean	0,030	12º	measur section	0,327
13º	data sampl	0,199	13º	data simul	0,030	13º	measur puriti	0,326
14º	effect mean	0,198	14º	sampl set	0,029	14º	defect measur	0,324
15º	sampl set	0,189	15º	effect mean uniform	0,028	15º	equival measur	0,323
16º	extern measur valid	0,184	16º	mean uniform	0,025	16º	measur result	0,321
17º	mean uniform	0,179	17º	cluster result	0,024	17º	inform measur	0,319
18º	class distribut	0,179	18º	class size	0,023	18º	cluster mean	0,310
19º	cluster mean	0,175	19º	cluster mean	0,022	19º	cluster result	0,303
20º	class data	0,172	20º	class distribut	0,019	20º	class cluster	0,283
21º	cluster measur	0,172	21º	measur properti	0,017	21º	cluster evalu	0,262
22º	extern valid	0,171	22º	cluster valid	0,016	22º	cluster data	0,261
23º	result valid	0,166	23º	measur normal	0,013	23º	cluster number	0,259
24º	measur normal	0,162	24º	defect measur	0,012	24º	data set	0,254
25º	class set	0,161	25º	cluster evalu	0,012	25º	cluster set	0,254

appears only in the ranking of the Kappa measure. The different interest measures used in this paper will be compared to verify which of them produce better results.

The fourth step consists of using the generated features to build a representation in the vector space model.

The bag-of-related-words avoids several problems mentioned in Section 2. For instance, the proposed approach does not analyze the whole dimensionality of the textual document collection. Instead, each document is analyzed individually. The number of words in each document can be orders of magnitude smaller than the number of words in the text document collection.

An interesting characteristic of the bag-of-related-words is that the setting of the number of words in the feature is not required. The rules will be generated until the values of support and some interest measure are higher than the thresholds informed by the user.

To take away the responsibility of the user of setting the minimum support value, and to allow the setting of this value according to each document, we proposed the following formula:

$$AutSup = \frac{GeneralMeanFreq}{NumberTrans} \qquad (1)$$

where $GeneralMeanFreq$ means the sum of the frequency of all the words divided by the number of different words, and $NumberTrans$ means the number of transactions.

The definition of the thresholds of other interest measures can be also set automatically. In this paper we used a threshold based on mean ($\overline{x}$) of the interest measure values of the extracted rules in a document. According to the thresholds chosen by the user during the process, the dimensionality can be much lower than the bag-of-words, and the application of feature selection methods is not necessary. However, as the features are represented in the vector space model, these methods can be applied as an additional task.

Moreover, the proposed approach is independent on i) the natural language processing, which is normally computationally expensive ii) the interference of domain specialists, and iii) knowledge base.

4. EXPERIMENTS AND RESULTS

Two experiments were carried out to analyze the feasibility to aid the topic hierarchy building using the bag-of-related-words and the bag-of-words. The first experiment was an objective evaluation of the quality of the hierarchical clustering. The second experiment was a subjective evaluation about the topics of the hierarchy.

We created 8 collections of proceedings from the ACM digital library for the experiments. Each collection has 5 classes and approximately 90 documents per class[4].

First we evaluated which way to map the document into transactions produced better results to the clustering task. We used only the frequent itemsets in this step of the evaluation so that the objective measures do not interfere in the results. We considerer six possibilities to map the textual document into transactions: sentences, paragraphs, and sliding windows of sizes 5, 10, 20, and 30. A set of minimum support thresholds was used for each way of mapping. Table 4 presents the minimum support thresholds used in the experiments and Figure 1 presents the number of features obtained. The values of the minimum supports were defined based on previous experiments. It can be noticed that most of the used minimum support made the approach obtain a lower number of features than the *bag-of-words* approach for most of the collections.

After evaluating which way of mapping produced better results, we focused on the better way of mapping and apply the interest measures mentioned in Section 3 to prune the number of features. For each interest measure, we also used a set of thresholds based on the number of generated features. We also considered the threshold based on mean ($\overline{x}$).

[4] http://sites.labic.icmc.usp.br:8088/ragero/Acm-Collection/

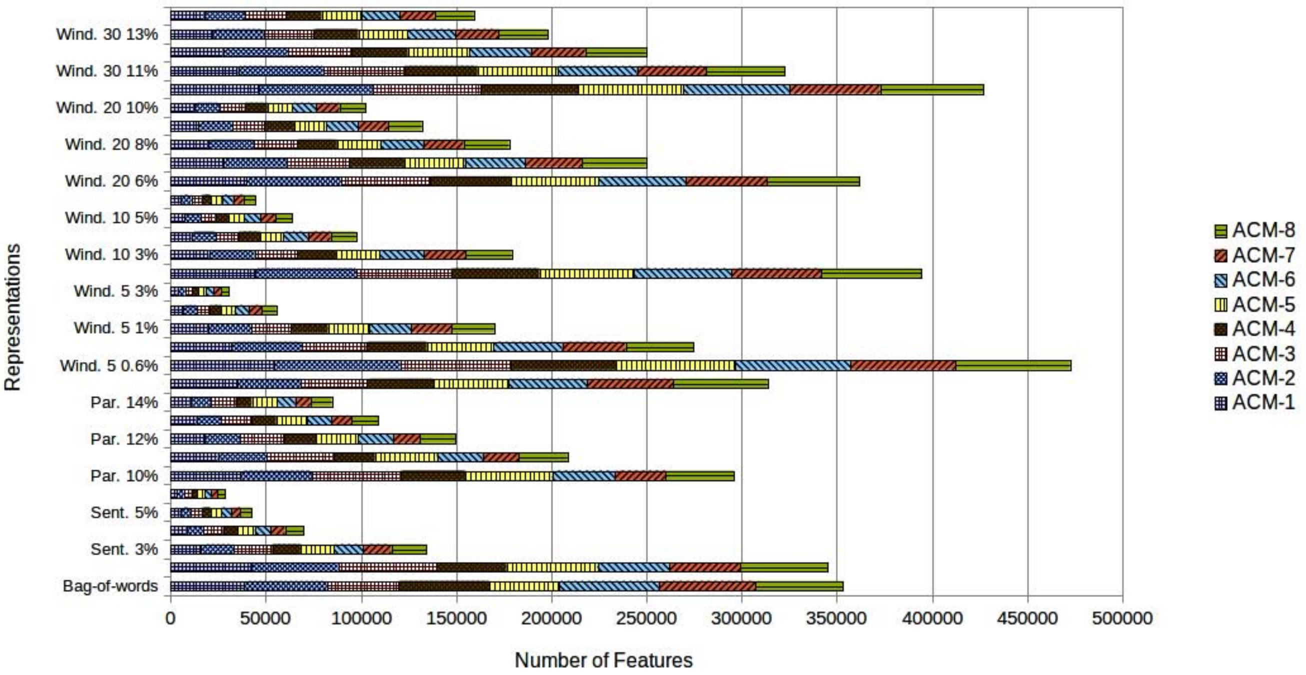


Figure 1: Number of features obtained by the different types of mapping and their thresholds.

Table 4: Minimum support values used in the experiments.

Type of Trans.	Min. Sup. Values
Sentence	2%, 3%, 4%, 5%, 6%
Paragraph	9%, 10%, 11%, 12%, 13%
Sliding Wind.: size 5	0.6%, 0.8%, 1%, 2%, 3%
Sliding Wind.: size 10	2%, 3%, 4%, 5%, 6%
Sliding Wind.: size 20	6%, 7%, 8%, 9%, 10%
Sliding Wind.: size 30	10%, 11%, 12%, 13%, 14%

Table 5 presents the used thresholds and Figure 2 shows the number of obtained features. As the measures Lift range from 0 to ∞, the standardization proposed in [11] was used.

Table 5: Thresholds of the interest measures used in the experiments.

Interest Measure	ACM	Reuters
Confidence	0,25; 0,50; $\overline{x}$	0,25; 0,50; $\overline{x}$
Lift	0,10; 0,20; $\overline{x}$	0,10; 0,20; $\overline{x}$
Yule's Q	0,50; 0,75; $\overline{x}$	0,50; 0,75; $\overline{x}$
Correlation	0,25; 0,50; $\overline{x}$	0,25; 0,50; $\overline{x}$
Mutual Information	0,005; 0,01; $\overline{x}$	0,01; 0,05; $\overline{x}$
Gini Index	0,005; 0,01; $\overline{x}$	0,01; 0,04; $\overline{x}$
Kappa	0,15; 0,20; $\overline{x}$	0,30; 0,50; $\overline{x}$
J-Measure	0,01; 0,02; $\overline{x}$	0,02; 0,05; $\overline{x}$

The variations of the bag-of-related-words will be represented between brackets. The type of mapping, the minimum support values, the interest measure, and the threshold of the interest measure will be presented inside the brackets. For instance, the representation that used the mapping of sentences with a minimum support value of 2%, and the Confidence measure with a threshold of 70%, will be represented by [Sent. 2% Confidence 70%].

The Pretext tool [16] was used to generate the bag-of-words representation.

4.1 Textual Document Clustering Evaluation

In the first experiment, the feasibility of the bag-of-related-words and the bag-of-words to build a hierarchical clustering was analyzed. The UPGMA (Unweighted Pair Group Method with Arithmetic Mean) algorithm was used to build an hierarchical clustering. This is an agglomerative hierarchical clustering algorithm of the type *average-link*. The UPGMA algorithm obtains results as good as the bisecting-k-means for the textual document hierarchical clustering. We decided to use the UPGMA because the results of the bisecting-k-means depends on the choices of the initial seeds to partition the clusters [23].

The metric used to compare the hierarchical clusters was the *FScore* measure. This measure analyzes the hierarchy group and compares it with the document categories. The *FScore* measure uses the following concepts [23]:

- L_r: represents a class r from the collection;
- n_r: represents the number of documents of a class r;
- S_i: represents a subgroup i of the obtained clustering;
- n_i: represents the number of documents from group i;
- n_{ri}: represents the number of documents from group Si that belong to the class L_r.

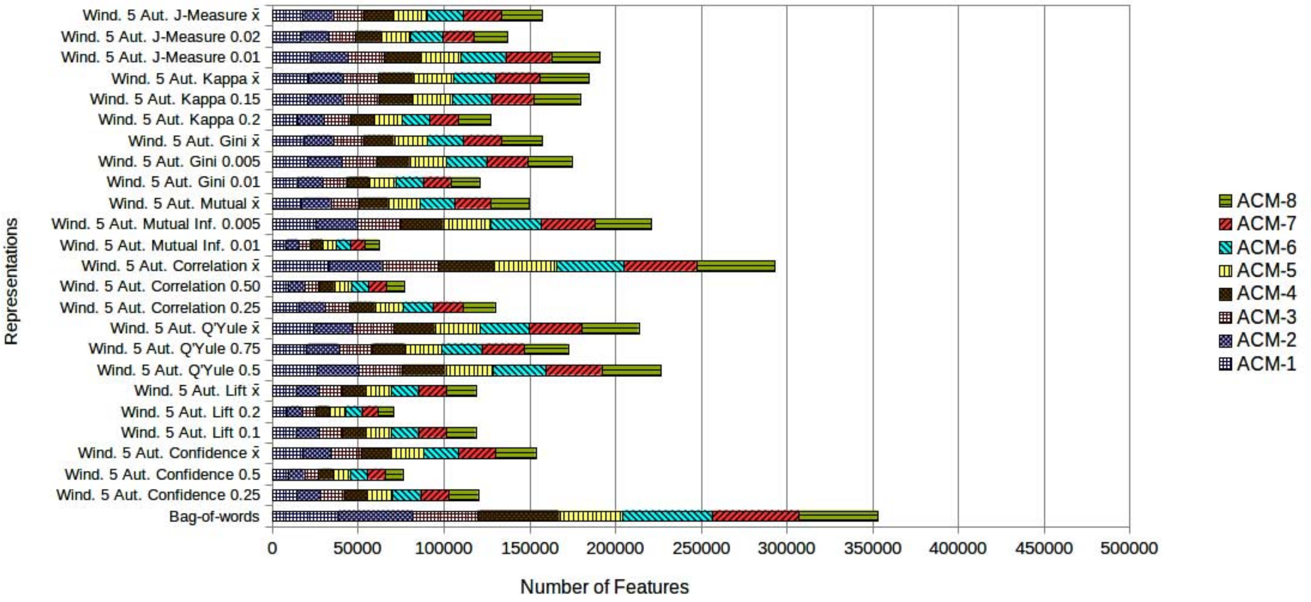


Figure 2: Number of features obtained by the representations using interest measures and their thresholds.

The *FScore* is an harmonic mean between precision (π) and recall (ρ) [2]. The precision and recall for a class L_r and a group S_i are the following:

$$\pi(L_r, S_i) = \frac{n_{ri}}{n_i} \tag{2}$$

$$\rho(L_r, S_i) = \frac{n_{ri}}{n_r} \tag{3}$$

After the *FScore* for each class and their respective groups were obtained, the *FScore* value for a class L_r is the biggest *FScore* value of that class for any group of the hierarchy T, that is:

$$F(L_r) = max_{S_i \in T} F(L_r, S_i) \tag{4}$$

The overall FScore value for the entire clustering solution is the sum of the *FScore* values for all the classes (c) weighted by the class size. Then, the *FScore* value for the clustering is:

$$FScore = \sum_{r=1}^{c} \frac{n_r}{n} F(L_r) \tag{5}$$

This measure ranges from $[0, 1]$, where 1 represents the maximal quality for the clustering.

Table 6 presents the representations that obtained the best *FScore* values for some ACM collection using only the frequent itemsets[5]. These results aim to evaluate which way to map the textual document into transactions can produce the better results. Using the frequent itemsets as features, the highest values of the *FScore* measure was obtained by the bag-of-related-words for most of the collections (7 of 8 collections). There were considerable differences in comparison with the bag-of-words for some collections, such as ACM-2, with a difference of 0.111, and ACM-7, with a difference of 0.130.

For some collections, as ACM-1 and ACM-3, the best results were obtained by the representations based on the bag-of-related-words [Wind. 5 3%] and [Wind. 10 5%]. The reduction in the number of feature was approximately 91% and 80% respectively in relation to the bag-of-words.

In general, representations based on the mapping of size 5 sliding window obtained the best results. The automatic minimum support setting was used in the mapping of size 5 sliding window, since it obtained the best results. Table 7 presents a comparison of the results obtained by the automatic support and the other used thresholds. It can be noticed that the use of automatic minimum support setting presented results as good as the manually set minimum support.

The Friedman test was applied considering the 8 ACM collections to verify how the variation of the minimum support values affected the results for the different types of mapping analyzed. There was significant statistical difference for the results obtained by the sentence mapping, where the lowest minimum support thresholds obtained better results than the highest minimum support values.

Then the best values of minimum support for each type of mapping were chosen to compare the several types of mapping and the bag-of-words representation. The representations chosen were: [Sent. 2%], [Par. 10%], [Wind. 5 2%], [Wind. 10 5%], [Wind. 20 8%], and [Wind. 30 11%]. None of the representations presented statistical significant differences. It must be highlighted that the representation [Wind. 5 2%] presented better results than the bag-of-words and reduced the number of features more than 80%.

[5] The complete results are available in `http://sites.labic.icmc.usp.br:8088/ragero/DocEng2011/docint.pdf`

Table 6: Results of the representations using frequent itemsets that obtained the highest value of the FScore measure for some ACM collection.

Representação	ACM-1	ACM-2	ACM-3	ACM-4	ACM-5	ACM-6	ACM-7	ACM-8
Bag-of-words	0,779	0,818	0,821	0,854	0,842	0,892	0,800	*0,867
Wind. 5 0,06%	0,771	0,820	0,863	0,917	0,868	*0,910	0,812	0,792
Wind. 5 0,08%	0,788	0,845	0,891	0,900	*0,881	0,893	0,803	0,813
Wind. 5 2%	0,806	0,912	0,843	0,909	0,829	0,880	*0,930	0,795
Wind. 5 3%	*0,813	0,915	0,804	0,886	0,822	0,841	0,897	0,749
Wind. 10 5%	0,761	0,924	*0,904	0,891	0,816	0,868	0,904	0,817
Wind. 20 7%	0,765	0,874	0,877	*0,921	0,833	0,849	0,895	0,768
Wind. 30 11%	0,752	*0,929	0,893	0,889	0,814	0,872	0,919	0,764

Table 7: Results of the representations using the mapping of size 5 sliding window.

Representation	ACM-1	ACM-2	ACM-3	ACM-4	ACM-5	ACM-6	ACM-7	ACM-8
Bag-of-words	0.779	0.818	0.821	0.854	0.842	0.892	0.800	*0.867
Wind. 5 Aut.	0.802	0.886	0.889	*0.917	0.868	0.891	0.823	0.821
Wind. 5 0.06%	0.771	0.820	0.863	*0.917	0.868	*0.910	0.812	0.792
Wind. 5 0.08%	0.788	0.845	0.891	0.900	*0.881	0.893	0.803	0.813
Wind. 5 1%	0.784	0.886	*0.900	0.913	0.848	0.895	0.850	0.820
Wind. 5 2%	0.806	0.912	0.843	0.909	0.829	0.880	*0.930	0.795
Wind. 5 3%	*0.813	*0.915	0.804	0.886	0.822	0.841	0.897	0.749

Table 8: Results of the representations using interest measures that obtained the highest value of the FScore measure for some ACM collection.

Representation	ACM-1	ACM-2	ACM-3	ACM-4	ACM-5	ACM-6	ACM-7	ACM-8
Bag-of-words	0.779	0.818	0.821	0.854	0.842	0.892	0.800	*0.867
Wind. 5 Aut. Conf. 0.5	*0.855	0.882	0.886	0.922	0.911	0.884	0.799	0.744
Wind. 5 Aut. Lift 0.1	0.799	*0.889	0.892	0.918	0.901	0.913	0.786	0.814
Wind. 5 Aut. Lift Aut	0.799	*0.889	0.892	0.918	0.901	0.913	0.786	0.814
Wind. 5 Aut. Q'Yule 0.75	0.796	0.872	0.904	*0.926	0.902	0.910	0.763	0.824
Wind. 5 Aut. Corr. 0.25	0.789	0.854	0.896	0.922	*0.915	0.883	0.783	0.801
Wind. 5 Aut. Corr. 0.50	0.831	0.877	0.883	0.922	0.909	*0.925	0.805	0.778
Wind. 5 Aut. M.I. 0.01	0.853	0.883	*0.908	0.920	0.904	0.923	0.814	0.765
Wind. 5 Aut. Kappa $\overline{x}$	0.778	0.875	0.872	0.918	0.905	0.897	*0.815	0.803

Interest measures were applied considering the mapping of size 5 sliding window, since it obtained the best results. We also decide to use the automatic minimum support setting since they obtained results as good as the other minimum support values. Table 8 presents the results of the representations that obtained the best results for some ACM collection.

The representations that used interest measures obtained better results than the bag-of-words for 7 of 8 ACM collections (ACM-1, ACM-2, ACM-3, ACM-4, ACM-5, ACM-6, ACM-7). The interest measures reduced the number of features and obtained better results than the base representation ([Wind. 5 Aut.]) for the collections ACM-1, ACM-2, ACM-3, ACM-4, ACM-5, ACM-6, and ACM-8. All the representations on Table 8 always reduced the number of features in the minimum of 26%. In some situation, 82% of the number of features were reduced.

The Friedman test was applied again to verify if the thresholds of the used interest measures affected the values of the *FScore*. There was only statistical significant difference for the Kappa measure, in which the threshold based on mean presented better results than the threshold 0.15. Thus, the threshold variations of the interest measures reduced the number of features and kept the quality of the results. Also there was no evidence that the thresholds defined manually were better than the thresholds based on mean. Then, we decided to compare the representations with interest measures and thresholds based on mean with the bag-of-words representation. Again, the Friedman test was applied and there were not statistical significant differences among the representations.

The obtained results demonstrated that the bag-of-related-words is appropriate for the textual document clustering. The highest *FScore* values was obtained by the bag-of-related-words for most of the ACM collections. It also can be highlighted that in most of the situations, the best results were obtained with a reduced number of features.

4.2 Topic Hierarchy Evaluation

In the second experiment, topic hierarchies were built and submitted to computer science domain experts. Three ACM collections for the subjective evaluation were used: ACM-3, ACM-4, and ACM-8. Table 9 presents the topics of these collections. The representation [Wind. 5 Aut. Correlation $\overline{x}$] was chosen to be compared with the bag-of-words and the Torch[6] tool was used to build the topic hierarchies.

Two hierarchies of each collection were given to 6 computer science domain experts and the representation used

[6] `http://sites.labic.icmc.usp.br/marcacini/ihtc/`

Table 9: Topics of the collections used for the subjective evaluation of the topic hierarchy.

Col.	Topic
ACM-3	Computer Architecture Education
	Architecture for Networking And Commu. Systems
	Privacy in the Electronic Society
	Software and Performance
	Web Information and Data Management
ACM-4	Embedded Networked Sensor Systems
	Research and Development in Info. Retrieval
	Parallel Algorithms and Architectures
	Volume Visualization
	Cross-Disciplinary Conf. on Web Accessibility
ACM-8	Mobile Ad Hoc Networking & Computing
	Knowledge Discovery and Data Mining
	Langu., Comp. and Tool Sup. for Embedded Systems
	Hypertext and Hypermedia
	Microarchitecture

to build the topic hierarchy was not specified. The following questions were evaluated:
1) Is it possible to identify clearly the topics of the collections based on the respective categories?
2) Does the browsing through the topic hierarchy conduced the user to a desired document set?

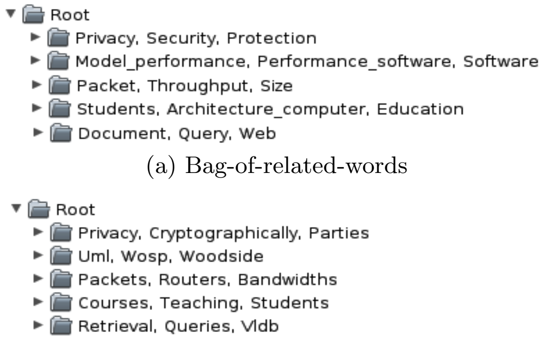
For both evaluations the used grades were:
0 - Nothing;
1 - Little;
2 - Fair;
3 - Good;
4 - Excellent.

Table 10 presents the means of the grades given to the domain experts for all the evaluated topic hierarchies. It can be noticed that in all the collections, the bag-of-related-words obtained better results than the bag-of-words.

Table 10: Mean of the results obtained in the subjective evaluation of the topic hierarchies for the ACM-3, ACM-4, and ACM-8 collection.

	ACM-3		ACM-4		ACM-8	
	BOW	BORW	BOW	BORW	BOW	BORW
Question 1	2.00	2.83	1.83	2.50	2.50	2.83
Question 2	2.00	2.66	2.00	2.50	2.16	2.66

To illustrate the topic hierarchies built using the bag-of-related-words and bag-of-words, Figures 3 and 4 presents the first level of the topic hierarchy for the ACM-3 and ACM-4 collections.



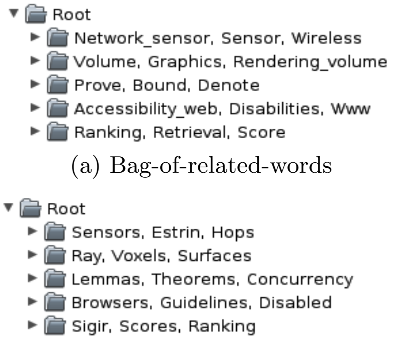
- Root
  - Privacy, Security, Protection
  - Model_performance, Performance_software, Software
  - Packet, Throughput, Size
  - Students, Architecture_computer, Education
  - Document, Query, Web

(a) Bag-of-related-words

- Root
  - Privacy, Cryptographically, Parties
  - Uml, Wosp, Woodside
  - Packets, Routers, Bandwidths
  - Courses, Teaching, Students
  - Retrieval, Queries, Vldb

(b) Bag-of-words

Figure 3: Example of the first level of the topic hierarchy using the bag-of-related-words and bag-of-words representations for the ACM-3 collection.



- Root
  - Network_sensor, Sensor, Wireless
  - Volume, Graphics, Rendering_volume
  - Prove, Bound, Denote
  - Accessibility_web, Disabilities, Www
  - Ranking, Retrieval, Score

(a) Bag-of-related-words

- Root
  - Sensors, Estrin, Hops
  - Ray, Voxels, Surfaces
  - Lemmas, Theorems, Concurrency
  - Browsers, Guidelines, Disabled
  - Sigir, Scores, Ranking

(b) Bag-of-words

Figure 4: Example of the first level of the topic hierarchy using the bag-of-related-words and bag-of-words representations for the ACM-4 collection.

5. CONCLUSION AND FUTURE WORKS

Topic hierarchies are useful for managing, searching, and browsing in large text collections. To allow the automatic topic hierarchy building, the textual document collection must be represented in an appropriate way. The proposed approach for textual document representation, bag-of-related-words, demonstrated to be useful for the topic hierarchy building.

The textual document clustering, used to create the hierarchical structure of the document collection, presented better results when performed using the bag-of-related-words representation. Also the topics of the obtained hierarchy using the bag-of-related-words were more understandable for the users than the topics obtained by the bag-of-words. It must be highlighted that the initial dimensionality of the proposed approach, even containing features composed by set of words, in most cases was much smaller than the bag-of-words with better results.

The mapping of a textual document into transactions using a size 5 sliding window presented the best results. The thresholds of interest measures used in the experiments in most cases do not present significant statistical difference. However, this setting is not trivial and can be time consuming for the user. An automatic threshold setting was used in this paper. The results obtained with the automatically set thresholds were equivalent or better than the manual threshold setting.

The analysis of each document individually, besides avoiding the entire dimensionality of the collection, allows the proposed approach be used in dynamic contexts, in which the analysis of the entire document collection to extract the features is unfeasible for each new document.

For future research, we are going to evaluate the pruning of redundant rules and redundant features to reduce even more the number of features without the need of a supervised feature selection method.

Acknowledgements

We would like to thank CNPq to the financial support.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB'94: International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] J. Blanchard, F. Guillet, R. Gras, and H. Briand. Using information-theoretic measures to assess association rule interestingness. In *ICDM'05: Internation Conference on Data Mining*, pages 66–73, 2005.

[4] M. F. Caropreso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Text databases & document management: theory & practice*, pages 78–102, 2001.

[5] A. L. C. Carvalho, E. S. Moura, and P. Calado. Using statistical features to find phrasal terms in text collections. *Journal of Information and Data Management*, 1(3):583–597, 2010.

[6] A. Doucet and H. Ahonen-Myka. Non-contiguous word sequences for information retrieval. In *MWE'04: Workshop on Multiword Expressions: Integrating Processing*, MWE'04, pages 88–95. Association for Computational Linguistics, 2004.

[7] L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3):9, 2006.

[8] F. Guillet and H. J. Hamilton, editors. *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*. Springer, 2007.

[9] V. Kashyap, C. Ramakrishnan, C. Thomas, and A. P. Sheth. Taxaminer: an experimentation framework for automated taxonomy bootstrapping. *International Journal of Web and Grid Services*, 1(2):240–266, 2005.

[10] Y. Lie, H. T. Loh, and W. G. Lu. Deriving taxonomy from documents at sentence level. In A. H. do Prado and E. Ferneda, editors, *Emerging Technologies of Text Mining: Techniques and Applications*, chapter 5, pages 99–119. Information Science Reference, 1 edition, 2007.

[11] P. D. McNicholas, T. B. Murphy, and M. O'Regan. Standardising the lift of an association rule. *Computational Statistics & Data Analysis*, 52(10):4712–4721, 2008.

[12] D. Mladenic and M. Grobelnik. Word sequences as features in text-learning. In *ERK'98: Electrotechnical and Computer Science Conference*, pages 145–148, 1998.

[13] M. F. Moura and S. O. Rezende. A simple method for labeling hierarchical document clusters. In *IASTED'10: International Conference on Artificial Intelligence and Applications (IAI 2010)*, pages 363–371, 2010.

[14] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[15] F. F. Santos, V. O. de Carvalho, and S. O. Rezende. Selecting candidate labels for hierarchical document clusters using association rules. In Springer-Verlag, editor, *MICAI'10: Mexican International Conference on Artificial Intelligence*, 2010.

[16] M. V. B. Soares, R. C. Prati, and M. C. Monard. PreTexT II: Descrição da reestruturação da ferramenta de pré-processamento de textos. Technical Report 333, ICMC-USP, 2008.

[17] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546, 2002.

[18] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *ACM SIGKDD'2002: International Conferenceon Knowledge Discovery and Data Mining*, pages 32–41. ACM, 2002.

[19] R. Tesar, V. Strnad, K. Jezek, and M. Poesio. Extending the single words-based document model: a comparison of bigrams and 2-itemsets. In *DocEng'06: ACM Symposium on Document Engineering*, pages 138–146, 2006.

[20] J. Wu, H. Xiong, and J. Chen. Adapting the right measures for k-means clustering. In *SIGKDD'09: Proceeding of the International Conference on Knowledge Discovery and Data Mining*, pages 877–886. ACM, 2009.

[21] Z. Yang, L. Zhang, J. Yan, and Z. Li. Using association features to enhance the performance of naïve bayes text classifier. In *ICCIMA '03: International Conference on Computational Intelligence and Multimedia Applications*, page 336. IEEE Computer Society, 2003.

[22] X. Zhang and X. Zhu. A new type of feature - loose n-gram feature in text categorization. In *IbPRIA'07: Iberian Conference on Pattern Recognition and Image Analysis*, pages 378–385. Springer, 2007.

[23] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: International Conference on Information and Knowledge Management*, pages 515–524. ACM Press, 2002.