Proceedings of International Workshop on Data Mining and Artificial Intelligence (DMAI' 08)
24 December, 2008, Khulna, Bangladesh

Making Good Choices of Non-Redundant N-gram Words

Maria Fernanda Moura
Embrapa Informática Agropecuária
PO Box 6041, Campinas, SP, Brazil
fernanda@cnptia.embrapa.br

Bruno Magalhães Nogueira
Merley da Silva Conrado
Fabiano Fernandes dos Santos
Solange Oliveira Rezende
University of São Paulo
Computer and Mathematics Science Institute
PO Box 668, São Carlos, SP, Brasil
{brunomn,merleyc,fabianof,solange}@icmc.usp.br

Abstract

A new complete proposal to solve the problem of automatically selecting good and non redundant n-gram words as attributes for textual data is proposed. Generally, the use of n-gram words is required to improve the subjective interpretability of a text mining task, with $n \geq 2$. In these cases, the n-gram words are statistically generated and selected, which always implies in redundancy. The proposed method eliminates only the redundancies. This can be observed by the results of classifiers over the original and the non redundant data sets, because, there is not a decrease in the categorization effectiveness. Additionally, the method is useful for any kind of machine learning process applied to a text mining task.

1. Introduction

The amount of information in the digital universe corresponded to 161 exabytes according to 2006 estimates, which was about 3 million times the information in all books already written. In the period from 2006 to 2010, an increase in the digital universe has been estimated from 161 exabytes to 988 exabytes [6]. Facing this scenario, the human capability of analyzing and understanding those data has been exceeded. Therefore, much of the information and knowledge, potentially useful, can be wasted, hidden in data bases spread throughout the world. As a consequence, the need for developing new tools and knowledge extraction techniques from storage data is becoming more and more indispensable [5].

The Text Mining (TM) process, using computational techniques of knowledge extraction, acts in this scenario as a transformer agent, extracting from this enormous quantity of textual data some useful knowledge that can be used as a competitive advantage or as a support to decision making. This process can be seen as a particular case of Data Mining and it is composed by five steps: problem identification, pre-processing, pattern extraction, pos-processing and knowledge use. These steps can be instanced according to the process goals [18].

Frequently, the pre-processing is considered a step of minor importance, or less interesting than the others, due to the lack of technical glamor and the excess of manual tasks. Basically, this step aims at transforming the text collection into a useful form for the learning algorithms, involving tasks as treatment, cleaning and reduction of the data. Additionally, in this step, the most representative features in the collection must be considered, looking for the elimination of the irrelevant features. Nevertheless, the difficulties and challenges imposed by this nontrivial step result in the rich or poor quality of the analyzed data and, consequently, in the efficiency of the used learning algorithms and the confidence of the obtained results. Moreover, it is important to consider the explicit knowledge contained in the attributes in a text mining task, since the final results can be subjectively analyzed [20].

In this way, the probability of considering an attribute as a domain term in a text collection can be taken as a quality and discriminative property. Although many authors consider the dimensionality increase more harmful than beneficial, the attributes representation using single and compound words seems to be more easily interpreted and, consequently, it is more suitable to the users of the results of a text mining task ([14], [15]). When working with single and compounded words, found after a statistical analysis, there are some observed redundancies and a lot of compounded words that can be discarded. Some initiatives to discard redundancies can be found in the work of Zhang and Zhu

1-4244-2136-7/08/$20.00 ©2008 IEEE

[21] and of Dejun and Maosong [3]. According to Dejun and Maosong there is a kind of redundancy arising from feature overlapping in Chinese character n-gram and they have proposed a model to avoid the redundancies that they believed to be language independent, in which the overlapping is estimated and then absorbed by the attribute model; besides, the model is not confined to supervised learning. In the proposal of Zhang and Zhu, a window length between words that form a n-gram is estimated in a model to set a discriminative value to the n-gram words, based on the document frequency of the found n-gram words and in the class label value; consequently, it is better applied to supervised learning, but it is an interesting solution to remove words among the goal word grams.

In this work, a new complete proposal to solve the problem of automatically selecting good and non redundant n-gram words as attributes for textual data is detailed. The model is useful to supervised and unsupervised learning and it depends only on statistical tests over the attribute candidates. In the next section the methodology to select good and non redundant n-gram words is described. In the experiments and results, section 3, we can observe that the reduction of feature space did not sacrifice the categorization results. Finally, some the conclusions and future work are discussed in section 4.

2. Proposed Method

The general idea is to generate attributes and apply filters that have more reliable behavior for each kind of n-gram sequentially, using statistical methods only, so the whole process can be completely automated. First of all, there are some terms that has to be defined according to their use in this paper:

- **n-gram word**: the used n-gram word model removes the stopwords from the texts, identify simple words as tokens, apply a stemming process over the words and then combine all of them in a sequential way [13]. The stemming process was based on Porter's algorithm [17] and adapted to the Portuguese and Spanish languages, as well as English. For example, a n-gram word can be "artific", "artific-intelligenc", "intelligenc-conferenc" and "artific-intelligenc-conferenc".

- **stopwords**: always the habitual list of stopwords, as articles and other grammatical classes that are very common in the texts and can be discarded from the attribute set. The set of words discarded from a filter selection is also considered stopwords, and in this paper this set of discarded words is called **collection stopwords**, referring to the specific text collection used in the example processes.

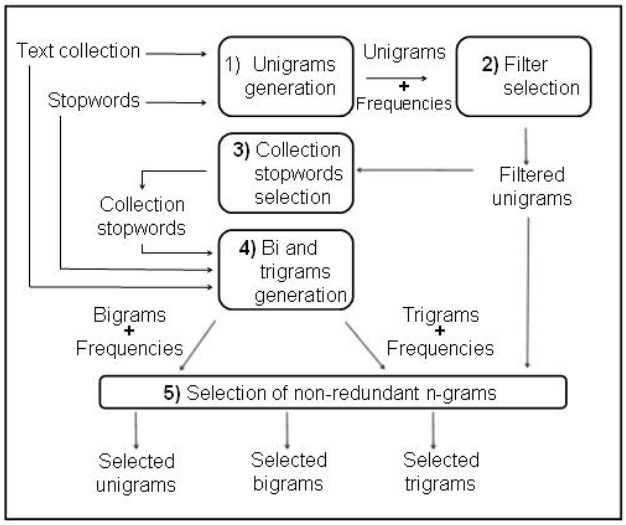

Figure 1. Steps of non-redundant n-gram word selection

- **attribute** or **term**: the two words refer to a selected n-gram word, that is, a n-gram word that is accepted as a valid attribute or a candidate attribute, because of its statistical discriminative potential.

The method is divided in five steps as illustrated in Fig.1. The text collection is supposed to be from a specific domain, because this method sees the text collection as a bag of n-gram words, so it is not able to treat polysemous words (or n-gram words) in different contexts; consequently this assertion decreases the probability of polisemy occurrence in the collection. For the first step, **unigram generation**, the text collection and the habitual stopword list are considered to generate only the unigram word set of candidate attributes and their frequencies are counted; that is all possible unigram word forms is in this set, for example, "artific", "intelligenc" and "conferenc".

In the second step, a **filter selection** is applied to the generated unigram word set. The filters are based on the observed frequencies and their variance. The most common used filter attribute is the Luhn´s method [11], that finds two cutoff points by ascending and plotting the attributes. So, two cutoff points are chosen close to the tendency curve inflexion points, considering the attributes which have very low or very high frequencies as irrelevant. Despite that, the elimination of low frequency terms is not a commonsense. For example, in the information retrieval area they have been favored, because of the wide use of the tf-idf indexer (tf-idf: term frequency - inverse document frequency) [19]. Some filters based on variance representativeness are also used: the Term Contribution (TC) [10] that can be defined as how much one specific term contributes to the simi-

larity among documents in a document collection; the Term Variance (TV) measure [9] uses the variance of all terms in the collection, attributing the highest scores to those terms that do not have low document frequency and have a non-uniform distribution through the collection; and, the Term Variance Quality (TVQ) [4] that is very similar to Term Variance and used to calculate the quality of a term through the total variance. Therefore, these filters provide an attribute ranking which implies in a cutoff choice. All of these filters are applicable for unsupervised attribute sets, because the most of interesting documents are unstructured and not classified. Besides, filters based on frequencies are more reliable to one gram sets, because they have a tendency on favoring terms that are not so frequent neither so rare. Experimentally it is easy to note that composed terms tend to have very low frequencies compared to simple terms (or unigram words). Moreover, the most interesting composed terms generally are the most frequent ones, although this observation does not guarantee that they can be considered domain terms or collocation candidates[12]. In this step, the filter that presents the best behavior is selected and the unigram word set selected by the filter is considered as a **valid attribute set**.

After obtaining the **filtered unigram word** set, in the third step, the **collection stopwords** are selected. They correspond to those single terms discarded from the text collection, that is, those words that do not help in the construction of the filtered unigram word set. The collection stopwords have to be discarded of the high order n-gram words because the interesting high order n-gram words are only those whose are formed by unigram words that are statistically discriminative. In this way, many of the possible n-gram words for the text collection have already been eliminated. In this work only **bigram and trigram words are being generated** in the fourth step so far.

Finally, there are three sets of attributes, where only the one gram set has already been evaluated by some statistical tests, and there are still a lot of redundant n-gram words. For example, in the text "*Textual data mining is data mining over textual data. Textual data are textual collections in several formats. Data mining has several definitions, but data are always data that are aimed at a special meaning and use.*" there are several possibilities of constructing n-gram words. Not all combinations of unigram words are interesting to discriminate the text or make any sense. In table 1 there are only the trigram, bigram and unigram words, presented in the text, that were subjectively considered to be useful and to have sense. In the **second column** there are the observed occurrence frequencies of each n-gram word in the text. As each low order n-gram word is also presented in a high order n-gram word, in the **third column** the number of occurrences was subtracted from the highest order n-gram word, in this first case from the trigram word; and, finally, in the **fourth column** the last number of frequencies for each n-gram word is subtracted from the last obtained frequencies for each bigram word in which it also occurs. As a result of this process, the n-gram words that have null frequency in the last column are discarded from the original set of attributes candidate, because their linear combinations of frequencies have resulted in higher order n-gram words. The n-gram words that have a non zero frequency are the non redundant. The systematization of these ideas occurs in the **selection of non redundant n-gram words**, fifth step illustrated in Fig. 1, and are described as follow.

Subjectively chosen n-grams	observed frequency	- 3gram occurrence	- 2gram occurrence
textual data mining	1		
data mining	3	3-1=2	
textual data	3	3-1=2	
textual collections	1	1-0=1	
several formats	1	1-0=1	
several definitions	1	1-0=1	
several meaning	1	1-0=1	
special meaning	1	1-0=1	
data	7	7-1=6	6-2-2-1=1
mining	3	3-1=2	2-2=0
textual	4	4-1=3	3-2-1=0
collections	1	1-0=1	1-1=0
several	2	2-0=2	2-1-1=0
formats	1	1-0=1	1-1=0
definitions	1	1-0=1	1-1=0
aimed	1	1-0=1	1-0=1
special	1	1-0=1	1-1=0
meaning	1	1-0=1	1-1=0
use	1	1-0=1	1-0=1

Table 1. Redundant selected n-gram words

First of all, the n-gram words, with $n = 3, 2, 1$, must be statistically selected as candidates for domain terms, instead of the subjectively selection. This can be carried out by using a set of tests for dependence among the unigram words of the generated n-gram in the whole text collection ([12], [1]). To simplify this process, the Yule´s Q test, a robust test used in decomposed contingency tables (for details see [2]) was chosen and incorporated to the method presented here. To reach all test results, a set of frequency measures has to be taken from each n-gram word.

Let $w_1w_2w_3$ be a trigram word and its observed frequency be f_{123}, it is also needed to obtain the observed frequency of the bigram word w_1w_2 in the first and second positions of all trigram words in the collection as $f_{12.}$, the observed frequency of w_2w_3 in the second and third positions as $f_{.23}$; and the frequencies of w_1 in the first position ($f_{1..}$) and w_3 in the third position ($f_{..3}$), of all obtained trigram words. The frequency of all trigram words in the collection ($f_{...}$) has also to be obtained.

In table 2, the frequency values for w_1 in the first position ($f_{1..}$) and w_2w_3 in the second position ($f_{.23}$) of the

trigram word $w_1w_2w_3$ (f_{123}) are illustrated, and, the frequency values of the n-gram words different from them in all the collection.

	w_2w_3	$!(w_2w_3)$	
w_1	f_{123}	$f_{1..} - f_{123}$	$f_{1..}$
$!(w_1)$	$f_{.23} - f_{123}$	$(f_{...} - f_{1..}) - (f_{.23} - f_{123})$	$f_{...} - f_{1..}$
	$f_{.23}$	$f_{...} - f_{.23}$	$f_{...}$

Table 2. w_1 in the first and w_2w_3 in the second and third positions of $w_1w_2w_3$

To test the hypothesis of the independence between w_1 in the first position and w_2w_3 in the second and third positions, using the Yule´s Q statistic, the following values must be obtained (for details see [2]):

$$d_1 = ((f_{...} - f_{1..}) - (f_{.23} - f_{123}))$$

$$\alpha_1 = \frac{f_{123} * d_1}{(f_{.23} - f_{123}) * (f_{1..} - f_{123})}$$

$$Q_1 = (\alpha_1 - 1)/(\alpha_1 + 1)$$

$$v_1 = \sqrt{\frac{1}{f_{123}} + \frac{1}{f_{1..} - f_{123}} + \frac{1}{f_{.23} - f_{123}} + \frac{1}{d_1}}$$

$$\sigma_{Q_1} = \frac{1}{2} * (1 - Q_1^2) * v_1$$

$$IC_{Q_1} = [Q_1 - 2 * \sigma_{Q_1}, Q_1 + 2 * \sigma_{Q_1}]$$

In table 3, the frequency values for w_1w_2 in the first and second positions ($f_{12.}$) and w_3 in the third position ($f_{..3}$) of the trigram word $w_1w_2w_3$ (f_{123}) are illustrated, and, the frequency values of the n-gram words different from them in all the collection.

	w_3	$!(w_3)$	
w_1w_2	f_{123}	$f_{12.} - f_{123}$	$f_{12.}$
$!(w_1w_2)$	$f_{..3} - f_{123}$	$(f_{...} - f_{12.}) - (f_{..3} - f_{123})$	$f_{...} - f_{12.}$
	$f_{..3}$	$f_{...} - f_{..3}$	$f_{...}$

Table 3. w_1w_2 in the first and second and w_3 in the third positions of $w_1w_2w_3$

So, to test the hypothesis of the independence between w_3 in the third position and w_1w_2 in the first and second positions, using the Yule´s Q statistic, the following values must be obtained (for details see [2]):

$$d_2 = ((f_{...} - f_{..3}) - (f_{12.} - f_{123}))$$

$$\alpha_2 = \frac{f_{123} * d_2}{(f_{12.} - f_{123}) * (f_{..3} - f_{123})}$$

$$Q_2 = (\alpha_2 - 1)/(\alpha_2 + 1)$$

$$v_2 = \sqrt{\frac{1}{f_{123}} + \frac{1}{f_{..3} - f_{123}} + \frac{1}{f_{12.} - f_{123}} + \frac{1}{d_2}}$$

$$\sigma_{Q_2} = \frac{1}{2} * (1 - Q_2^2) * v_2$$

$$IC_{Q_2} = [Q_2 - 2 * \sigma_{Q_2}, Q_2 + 2 * \sigma_{Q_2}]$$

If $0 \notin IC_{Q_1}$ and $0 \notin IC_{Q_2}$ then $w_1w_2w_3$ is a candidate to be a domain term collocation and is also selected as a trigram word attribute candidate. All selected trigram words, as attribute candidates, are ranked by their occurrence frequency, that is, by their f_{123}. As soon as all trigram words were tested, selected and ranked, all bigram and unigram words which frequencies form the trigrams by linear combinations are eliminated with the Algorithm 1, according to their observed frequencies: $O_{f_{ij}}$, $i, j = 1, 2, 3$ for bigram words; and O_{f_i}, $i = 1, 2, 3$ for unigram words.

It has to be highlighted that, only the best ranked candidate trigram words can be selected as final attributes. Additionally, for each of these trigram words there are some bigram or unigram words with frequency greater or equal to their frequencies. In this way, the redundancy is a selection criteria and it is used to eliminate itself.

Input: *TriGramList*: list of trigram words ordered by rank value
Output: *FinalAttributesList*: list of final attributes with updated frequencies and selected trigram words
for *all trigram* $w_aw_bw_c$ *in TriGramList* **do**
  select next best ranked trigram word, $w_1w_2w_3$;
  search for w_1w_2, with $O_{f_{12}} \geq O_{f_{123}}$;
  search for w_2w_3, with $O_{f_{23}} \geq O_{f_{123}}$;
  search for w_1, with $O_{f_1} \geq O_{f_{123}}$;
  search for w_2, with $O_{f_2} \geq O_{f_{123}}$;
  search for w_3, with $O_{f_3} \geq O_{f_{123}}$;
  subtracted $O_{f_{123}}$ from all above n-gram $O_{f_{ij}}$ and O_{f_i};
  add $w_1w_2w_3$ in *FinalAttributesList*;
end

Algorithm 1: Trigram words selection algorithm

After the trigram word selection, the same process and ideas are applied to the bigram words that were not eliminated yet. In this way, the bigram word candidates are selected using the Yule's Q test and they are ranked by their occurrence frequencies.

Let w_1w_2 be a bigram word and its observed frequency be f_{11}, it is also needed to obtain the observed frequency of the unigram word w_1 in the first position in all bigram words in the text collection as $f_{1.}$ and the observed frequency of w_2 in the second position as $f_{.1}$. The frequency of all bigram words in the collection, $f_{..}$, has also to be obtained; and all these values are illustrated in table 4.

To test the hypothesis of the independence between w_1 in the first position and w_2 in the second position, using

	w_2	$!(w_2)$	
w_1	f_{11}	$f_{1.} - f_{11}$	$f_{1.}$
$!(w_1)$	$f_{.1} - f_{11}$	$(f_{..} - f_{1.}) - (f_{.1} - f_{11})$	$f_{..} - f_{1.}$
	$f_{.1}$	$f_{...} - f_{..3}$	$f_{..}$

Table 4. w_1 in the first and w_2 in the second positions of w_1w_2

the Yule's Q statistic, the following values must be obtained (for details see [2]):

$$d = ((f_{..} - f_{1.}) - (f_{.1} - f_{11}))$$

$$\alpha = \frac{f_{11} * d}{(f_{1.} - f_{11}) * (f_{.1} - f_{11})}$$

$$Q = (\alpha - 1)/(\alpha + 1)$$

$$v = \sqrt{\frac{1}{f_{11}} + \frac{1}{f_{1.} - f_{11}} + \frac{1}{f_{.1} - f_{11}} + \frac{1}{d}}$$

$$\sigma_Q = \frac{1}{2} * (1 - Q^2) * v$$

$$IC_Q = [Q - 2 * \sigma_Q, Q_1 + 2 * \sigma_Q]$$

So, if $0 \notin IC_Q$ then w_1w_2 is candidate of a domain term collocation and the bigram word is also selected as an attribute candidate. All selected bigram words, or attribute candidates, are ranked by their occurrence frequency, f_{11}. As soon as all the remaining bigram words were tested, selected and ranked, all remaining unigram word which frequencies form the bigrams by linear combinations are eliminated, according to their observed frequencies: O_{f_i}, $i = 1, 2$; and following the Algorithm 2.

Input: *BiGramList*: list of the remaining bigram words ordered by rank value
Output: *FinalAttributesList*: list of final attributes with updated frequencies and selected bigram words
for *all bigram* w_aw_b *in BiGramList* **do**
  select next best ranked bigram, w_1w_2 with remaining frequency $O_{f_{11}}$;
  search for w_1, with $O_{f_1} \geq O_{f_{11}}$;
  search for w_2, with $O_{f_2} \geq O_{f_{11}}$;
  subtracted $O_{f_{11}}$ from all above unigram O_{f_1} and O_{f_2};
  add w_1w_2 in *FinalAttributesList*;
end

Algorithm 2: Bigram words selection algorithm

It has to be highlighted that, only the best ranked bigram word candidates, for which there are some unigram words with frequency greater or equal their frequency, can be selected as final attributes. Note that, the redundancy is a selection criteria and it is used to eliminate itself.

Again, it has to be highlighted that, only the best ranked candidate bigram words can be selected as final attributes. Additionally, for each of these bigrams words there are some unigram words with frequency greater or equal to their frequencies. In this way, the redundancy is a selection criteria and it is used to eliminate itself.

Moreover, it is easy to see that the two algorithms are almost the same indicating that the process can be expanded to any number n of n-gram words, if desired.

Besides, the evaluation of the method is available through an implementation of some prototypes (developed in C) to test, rank and select the candidate attribute sets and the final attribute set. In this way it is possible to obtain the candidate ranked attribute lists and the final selected attribute sets, in order to compare them. Although the best comparison should be a subjective evaluation carried out with information specialists or domain specialists, because of the data and the attribute nature, it is not an easy task over big text collections. Even a small text collection frequently results in a huge attribute set for human judgment.

To perform a non subjective evaluation of the preprocessing results and obtain a validation of the generated attribute set, some labeled document collections are used within a supervised learning process and evaluation. In this way, all the described steps are applied to a labeled text collection, but considering it as non labeled. Two attribute sets are constructed to each text collection:

1. **ranked attributes**: these are the attributes selected only by ranking the generated bigram and trigram words, using first the independence test explained and then the filter to the generated unigram words.

2. **non redundant selected attributes**: applying all the steps explained before to all generated attribute candidates.

The independence test applied to rank and select the n-gram words is the same in the two attribute sets, because the goal is to compare the attribute sets only at the presence or absence of redundancy, avoiding any other possible effect.

Then an attribute-value matrix is constructed for each attribute set, where, each row vector represents a document and each column an attribute; the cells correspond to the occurrence frequency of the attribute in the document; and finally the last column corresponds to the codified label. So, to validate the results, for each obtained attribute set, two classification models are constructed using two largely known classification algorithms: C4.5 decision trees and Support Vector Machines (SVM). Both of them were chosen due to the fact that they can face well sparse attribute matrices, besides SVM is able to generalize even in the presence of many redundant features [8]. Additionally, to estimate the classifiers accuracy rate, the 10-fold cross validation process is used. Finally, the accuracy of the ob-

tained classifiers at each attribute set are compared using the t-student test.

3. Some Experimental Results

Three data sets from different domains and languages were selected. The first is a collection of complete articles from the Instituto Fábrica do Milênio (**IFM**) [7]; that is a Brazilian organization whose actions are focused on the search for manufacturing solutions for the needs of industry. This data set is composed of 614 documents in the Portuguese language, divided into 4 classes, with 291 documents in the majority class. The second document collection is the Case Based Reasoning- Inductive Logic Programming - Information Retrieval - Sonification (CBR-ILP-IR-SON) data set [1] composed of 681 document abstracts in the English language, classified according to 4 classes with 276 documents in the majority class; it is a scientific production collection; and, in this paper that base is referred as **CIIS**. The third text collection is a set of complete articles, in English, divided in five sub topics of Artificial Intelligence: Agents & Multiagents, Fuzzy Logic, Machine Learning, Planning & Scheduling e Robotics[2]; in this paper that data base is referred as **IA.**

A random selection process was used to choose balanced samples for the three text collections, obtaining approximately fifty documents per class in each collection. In this way, effects due to the unbalanced classes are eliminated. In table 5 there are the number of documents, number of generated unigram words for each collection, the number of filtered unigram words for each collection. The chosen filter was the term contribution, TC [10], taking only 10% of the ranked unigram word attributes due to some previous experiments. In these experiments it was observed that filters based on the term contribution to the explained variance select the best attribute sets with smaller sets than the other techniques [10] and some other experiments suggested that only 10% of the ranked attributes produces results as realiable as the bigger sets [16]. In the same table, the last two columns refer to the number of generated bigram and trigram words after the unigram word filtering and the collection stopwords selection.

Taking the attribute candidates corresponding to the three last columns of table 5, the fifth step of the selection of non redundant n-gram word attributes is applied. The implemented prototypes were adapted to produce the non redundant attribute sets and the only ranked attribute sets. In this way, two attribute sets were generated for each text collection; the number of non redundant (NR) selected attributes, generated in the fifth process step, is in Table 6 and the number of ranked (R) attributes, that is with the redundancies, are in Table 7.

[1] http://infoserver.lcad.icmc.usp.br/infovis2/PEx

[2] http://labic.icmc.usp.br/projects

Text Collection	**#docs**	**#gen. 1gram**	**#filt. 1gram**	**#gen. 2gram**	**#gen. 3gram**
IA	250	42753	4276	87297	30326
CIIS	200	10810	1081	17940	17944
IFM	193	20795	2080	82325	84200

Table 5. Results from first,second, third and fourth process steps

Text collection	**#NR 1gram**	**#NR 2gram**	**#NR 3gram**
IA	4264	26270	24303
CIIS	1065	1379	1831
IFM	1984	7043	42741

Table 6. Non redundant n-gram words

Text collection	#R 1gram	#R 2gram	#R 3gram
IA	4276	53595	30060
CIIS	1081	4284	2811
IFM	2080	48775	82354

Table 7. Ranked n-gram words

To each of the attribute subsets of each text collection, the two classifiers, C4.5 and SVM, were applied and their accuracy rate was estimated using 10-fold cross validation. It is important to highlight that the classifiers are not being compared; the focus is on comparing how the different attribute sets react under a same evaluation process.

Text collection	Ranked $acc \pm se$	NonRed $acc \pm se$	t-test $p - value$
IA	90 ± 5.42	89.6 ± 7.11	0.3838
CIIS	87 ± 9.19	87 ± 9.19	0.3880
IFM	70.45 ± 8.28	75.63 ± 7.88	0.1385

Table 8. $Classifier\ C4.5\ -\ H_0 : \overline{acc_r} = \overline{acc_{nr}}$

Text collection	Ranked $acc \pm se$	NonRed $acc \pm se$	t-test $p - value$
IA	$79.6 \pm 8,1$	82 ± 7.12	0.2968
CIIS	90 ± 8.82	90.5 ± 6.85	0.3837
IFM	59.97 ± 10.24	68.84 ± 11.74	0.0835

Table 9. $Classifier\ SVM\ -\ H_0 : \overline{acc_r} = \overline{acc_{nr}}$

In tables 8 and table 9, it is possible to see the evaluation results along with the estimated accuracies and their standard deviation for each classifier and for each attribute set. The attribute sets are ranked (Ranked) and non redundant (NonRed). At the last columns there are the results for the

t-student test for the hypothesis of the equality of the accuracy means ($H_0 : \overline{acc_r} = \overline{acc_{nr}}$), where: $\overline{acc_r}$ and $(se)_r$, the accuracy mean and the standard deviation for the ranked attribute set, and $\overline{acc_{nr}}$ and $(se)_{nr}$ the accuracy mean and standard deviation for non redundant attribute set. It was calculated using equation 1 to obtain a t_{value} and then calculating the cumulative probability of this value, under a t-student distribution. If the cumulative probability value is the interval of $[0.025, 0.975]$, the hypothesis of the statistical equality of the accuracies is accepted. This interval corresponds to the values of the t-student table for nine degrees of freedom, because the accuracies have been calculated from a 10-fold cross experiment.

$$t_{value} = \frac{\overline{acc_r} - \overline{acc_{nr}}}{\sqrt{\frac{(se)_r^2}{10} + \frac{(se)_{nr}^2}{10}}} \quad (1)$$

At a first glance, focusing only on the performance, for each classifier we have the feeling that the feature redundancy removal was better, especially because of the range of the standard deviations. But after applying the t-test, the equality hypothesis was accepted for all data sets and classifiers. In this way, we can observe that the redundancy removal did not sacrifice the categorization effectiveness. Besides, as the accuracy estimates were good, we believe the attribute selection was also good.

4. Conclusions

A new complete proposal to solve the problem of automatically selecting good and non redundant n-gram words as attributes for textual data is proposed. The method is based on the assumption that attribute redundancy is always present when using n-gram words, for $n \geq 2$, due to the stochastic nature of them.

The proposed method was shown to reach the main goals: selecting good n-gram words and eliminating only the redundancies. Firstly, the accuracy estimates of the classifiers were good, what is an indication of a good attribute selection. Moreover, the categorization effectiveness has been not decreased after the redundancy removing, which indicates that only redundant attributes had been eliminated.

Since the method does not depend on class labels, it can be used in unsupervised or supervised learning. We believe that for supervised learning it is better to change the filtering process used in the second step, because an appropriated filter could produce a better unigram word selection.

The proposed method is completely dependent only on statistical tests over a bag of n-gram words instead of contiguous character n-grams, as on the method of Dejun and Maosong. Besides, Dejun and Maosong believed their method would be language independent; and the proposed method is language independent after the stemming process, as we could observe in the experiments with text collections in two different languages (Portuguese and English).

Moreover, this method uses the collection stopwords, as defined here, to remove undesirable words among the goal n-gram words; in this way, there is no need of estimating a window length between the words, as Zhang and Zhu had proposed. The proposed method can be classified closer to the method of Zhang and Zhu, because we believe that the removal of the collection stopwords also causes a semantic information improvement similar to the window length estimation.

As future work, we intend to realize subjective analysis over the generated set of attributes, along with domain specialists, to evaluate the semantic improvement. Besides, it is necessary to expand the algorithms, and the implemented prototype, to deal with any number of n in the n-gram word model.

References

[1] S. Banerjee and T. Pedersen. The design, implementation, and use of the ngram statistics package. In *CICLing*, pages 370–381, 2003.

[2] Y. Bishop, S. E. Fienberg, and P. H. Holland. *Discrete Multivariate Analisys*. MIT Press, 1984.

[3] X. Dejun and S. Maosong. Raising high-degree overlapped character bigrams into trigrams for dimensionality reduction in chinese text categorization. In A. F. Gelbukh, editor, *CICLing*, volume 2945 of *Lecture Notes in Computer Science*, pages 584–595. Springer, 2004.

[4] I. Dhillon, J. Kogan, and C. Nicholas. Feature selection and document clustering. In M. W. Berry, editor, *Survey of Text Mining*, pages 73–100. Springer, 2003.

[5] N. F. F. Ebecken, M. C. S. Lopes, and M. C. d. Aragão. Mineração de textos. In S. O. Rezende, editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 13, pages 337–364. Manole, 1 edition, 2003.

[6] J. F. Gantz, D. Reinsel, C. Chute, W. Schlichting, J. McArthur, S. Minton, I. Xheneti, A. Toncheva, and A. Manfrediz. The expanding digital universe: A forecast of worldwide information growth through 2010. *External Publication of IDC (Analyse the Future) Information and Data*, pages 1–21, March 2007.

[7] IFM. IFM - Instituto Fábrica do Milênio, 2007. Disponível em: http://www.ifm.org.br/ [29/08/2008].

[8] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In C. Nedellec and C. Rouveirol, editors, *ECML*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.

[9] L. Liu, J. Kang, J. Yu, and Z. Wang. A comparative study on unsupervised feature selection methods for text clustering. *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference on*, pages 597–601, 30 Oct.-1 Nov. 2005.

[10] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma. An evaluation on feature selection for text clustering. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 488–495. AAAI Press, 2003.

[11] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal os Research and Development*, 2(2):159–165, 1958.

[12] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[13] E. T. Matsubara, C. A. Martins, and M. C. Monard. Pretext: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. Technical Report 209, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos, 2003.

[14] Y. Miao, V. Keselj, and E. E. Milios. Document clustering using character n-grams: a comparative evaluation with term-based and word-based clustering. In O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury, and W. Teiken, editors, *CIKM*, pages 357–358. ACM, 2005.

[15] M. F. Moura and S. O. Rezende. Choosing a hierarchical cluster labelling method for a specific domain document collection. In *EPIA- Encontro Portugues de Inteligência Artificial, 2007, Guimarães, Portugual. New Trends in Artificial Intelligence. Lisboa, Portugual: APPIA - Associação Portuguesa para Inteligência Artificial*, pages 812–823, 2007.

[16] B. M. Nogueira, M. F. Moura, M. S. Conrado, and S. O. Rezende. Avaliação de métodos não-supervisionados de seleção de atributos para mineração de textos. In *Proceedings of the First Workshop on Web and Text Intelligence (WTI) - Nineteenth Brazilian Symposium on Artificial Intelligence (SBIA)*, page To appear, 2008.

[17] M. Porter. An algorithm for suffixing stripping. *Program*, 14(3):130–137, July 1980.

[18] S. O. Rezende, J. B. Pugliesi, E. A. Melanda, and M. F. Paula. Mineração de dados. In S. O. Rezende, editor, *Sistemas Inteligentes: Fundamentos e Aplicações*, chapter 12, pages 307–335. Manole, 1 edition, 2003.

[19] G. Salton and C. S. Yang. On the specification of term values in automatic indexing. 1973.

[20] S. M. Weiss, N. Indurkhya, T. Zhang, and F. J. Damerau. *Text Mining - Predictive Methods for Analyzing Unstructured Information*. Springer Science+Business Media, Inc., March 2005. ISBN 0-387-95433-3.

[21] X. Zhang and X. Zhu. A new type of feature - loose n-gram feature in text categorization. In J. Martí, J.-M. Benedí, A. M. Mendonça, and J. Serrat, editors, *IbPRIA (1)*, volume 4477 of *Lecture Notes in Computer Science*, pages 378–385. Springer, 2007.