An Analysis of Constructed Categories for Textual Classification using Fuzzy Similarity and Agglomerative Hierarchical Methods

Marcus V. C. Guelpeli, Ana Cristina Bicharra Garcia and Flavia Cristina Bernardini

Abstract Ambiguity is a challenge faced by systems that handle natural language. To assuage the issue of linguistic ambiguities found in text classification, this work proposes a text categorizer using the methodology of Fuzzy Similarity. The clustering algorithms Stars and Cliques are adopted in the Agglomerative Hierarchical method and they identify the groups of texts by specifying some type of relationship rule to create categories based on the similarity analysis of the textual terms. The proposal is based on the methodology suggested, categories can be created from the analysis of the degree of similarity of the texts to be classified, without needing to determine the number of initial categories. The combination of techniques proposed in the categorizer's steps brought satisfactory results, proving to be efficient in textual classification.

Key words: Text Mining, Fuzzy Similarity, Agglomerative Hierarchical Method and Similarity Matrix

Marcus V. C. Guelpeli and Ana Cristina Bicharra Garcia
Departamento de Ciência da Computação
Instituto de Computação – IC
Universidade Federal Fluminense – UFF
Rua Passo da Pátria 156 – Bloco E – 3º andar
São Domingos – Niterói – RJ CEP: 24210-240
{mguelpeli,bicharra}@ic.uff.br

Flavia Cristina Bernardini
Departamento de Ciência e Tecnologia — RCT
Pólo Universitário de Rio das Ostras — PURO
Universidade Federal Fluminense — UFF
Rua Recife, s/n – Jardim Bela Vista – Rio das Ostras – RJ CEP: 28890-000
fcbernardini@vm.uff.br

1 Introduction

The access to means of information distribution is becoming easier day by day. Motivated by the great availability of computer resources and the ease of exchanging and storing information, institutions in the most diverse fields have produced and electronically stored a large amount of data. In light of this possibility, companies have started making their products available by these means of distribution, expanding their markets globally and maximizing profits. Until a short time ago, this fact was not seen as a competitive advantage or a support tool for decision-making with indicators of successes and failures. As such, the amount of information is currently very great and continues to grow every minute. As well as being large, the information is set up in a disorganized and non-standardized manner, making it difficult to locate and to access. For [33], more than 80% of the information is currently found in a textual format. These textual documents are released on the web on a daily level, creating large collections of information, such as: a variety of reports, product specifications, error reports and software warning messages, summaries, notes, electronic mail, a multitude of documents (newsletters, newspapers, magazines, etc.) and all sorts of textual electronic publications (virtual libraries, a variety of document collections, etc) [12].

One of the biggest problems in accessing these types of information consists in correctly identifying the subject of any given document. This identification, conducted for the purpose of indexation, is done manually by people, which leads to delay problems or imprecise indexations. Another problem encountered in this area is adapting the automatic systems to, based on words from the text, select a set of terms that is representative of the desired concept. People find it relatively easy to infer concepts from words in documents, because they possess a reasonable knowledge of grammar as well as knowledge of the world around them, which, in the literature, is also known as background knowledge. In contrast to humans, automatic systems do not have this natural ability and, yet, the language used to recover information has to be closer to the natural language. This language, which is less deterministic, more flexible and open, offers the user the possibility of formulating questions with great ease, so that they can locate the most relevant documents. However, language's semantic wealth imposes a few limitations to this type of categorization.

Having discussed some of the most decisive concerns in this area, most of which are related to the large amount of available information, it can be concluded that new means of access and manipulation of large quantities of textual information should be created. For example, the study conducted by [24] cites two main problems that result from the overload of information: one is related to the location of relevant information and the other concerns the knowledge identification and extraction present in the relevant information that was found. To identify the relevant information, it is often necessary to spend hours in front of a search engine. After having identified the relevant information, it is generally not found in isolation but, rather, accompanied by many other pieces of information or spread in a series of

documents, making it necessary to analyze the content matter of the information, then filter or extract the data that is actually important.

At present, there is an emerging field, called Textual Data Analysis [24], that is concerned with studying and solving these two previously cited phases. Another field is called Knowledge Discovery from Texts, as described in [9, 33, 27, 18]. Both fields involve the process of recovering, filtering, manipulating and summarizing the knowledge extracted from large sources of textual information and presenting it to the final user by making use of a variety of resources, which usually differ from the originals. Hence, it is important that the analysis and processing mechanisms focus on this type of information that is contained in documents. Computational methods that automatically classify the available textual documents should be used in order to recovery information with greater speed and faithfulness (when it comes to the content matter of the texts), so that they can be useful to the decision-making process within organizations. There are a number of systems aimed at making systemic information storage and processing both socially and economically rational and profitable. Some methodologies have contributed to the appearance of computational systems that are capable of acquiring new knowledge, new abilities and new ways of organizing the existing knowledge [22].

Text Mining (TM) has been making it possible to transform this large volume of information, which is generally non-structured, into useful knowledge, which is often innovative, for the companies. Its use allows people to extract knowledge from non-structured brute textual information, providing elements of support to Knowledge Management, which, in turn, is the way of reorganizing how knowledge is created, used, shared, stored and evaluated. In terms of technology, TM supports knowledge management by transforming the content of information repositories into knowledge that can be analyzed and shared by the organization [34].

TM is a field of technological research whose main goal is to search for patterns, trends and regularity in texts written in natural language. It is normally involved with the process of extracting interesting and non-trivial information from non-structured texts. In this way, the aim is to transform implicit knowledge into explicit knowledge [8]. The process of Text Mining was inspired by the process of Data Mining, which consists of "non-trivial extraction of implicit information that is previously unknown and potentially useful data" [10]. For [5] this is called Text Data Mining. It is in fact a relatively new interdisciplinary field that encompasses: Natural Language Processing, in particular Computational Linguistics, Machine Learning, Information Recovery, Data Mining, Statistics and Information Visualization. For [13], TM is the result of the symbiosis of these fields. Applying a process of TM may have many purposes: creating summaries; clusterization (grouping texts according to similarities in their content matter); identifying languages; extracting terms; text categorization; managing electronic mail, managing documents and research and market investigation.

The focus of this work is to use techniques of text clusterization to categorize textual documents. Clusterization techniques are used when the classes in the elements of the available domain are unknown and, hence, one is looking to automatically separate the elements into groups by some affinity criterion or similarity. Clusteri-

zation aids in the process of uncovering in-text knowledge, thereby facilitating the identification of patterns in the classes.

The aim of this work is to propose a categorizer by using fuzzy similarity to improve the issue of linguistic ambiguities found in text classification and to use the agglomerative hierarchical method to create categories from the similarity analysis of textual terms. This is based on the hypothesis that categories can be created from the suggested methodology. In other words, the degree of similarity of the texts to be categorized improves the quality of the cluster representation, which increases their identification capacity, as well as facilitates the comprehension of the resulting clusters. The Eureka categorizer [37, 39, 38] groups the text in clusters according to the similarity among the words that compose each sentende. Results using Eureka categorizer is used to compare with the results obtained with categorizer proposed in this work. Our experiments were conducted using Temário, RSS_Terra and Reuters corpora.

The paper is organized as follows. In section 2, we introduce the theoretical concepts of Fuzzy Similarity. Section 3 handles clusterization methods, especially the hierarchical ones, which are the focus of this work, as well as the algorithms used (Stars and Cliques). In Section 4 a method that uses fuzzy logic with a relative frequency calculation for the selection of characteristics is proposed in order to obtain the similarity matrix. In Section 5 we discuss the results that were obtained with the suggested categorizer, which are compared to results obtained with other categorizers in the literature. Finally, Section 6 presents the conclusions drawn about the proposed method and future works.

2 Fuzzy Similarity

Ambiguity is the greatest challenge that systems dealing with natural language have to face. Identifying the real meaning of a given word can be so complicated that sometimes the only way to do so is to ask the user. In the process of choosing a more adequate alternative to the mathematical treatment with regards to questions formulated in natural language, the use of fuzzy logic comes with a great advantage, because conventional logic presents some difficulties when it comes to representing abstract concepts. In conventional, or Boolean, logic, which is commonly used in computing, only two possible values are determined: true (1) or false (0). This logic is not ideal for systems that deal with natural language, since it is impossible to faithfully cover all of the representations of the linguistic context. These systems are based only on right or wrong, yes or no; that is, in only two values to represent an extremely complex world.

Fuzzy logic, on the other hand, is based on the theory of fuzzy sets, whose concepts and principles were first introduced by [40, 41]. Fuzzy logic is multivalued, meaning that there is a set of possible values. Hence, fuzzy logic can be defined as a logic that supports the approximate modes of reasoning, instead of exact ones. The mathematical treatment of fuzzy logic is more appropriate for dealing with impre-

cise information that is generally employed during human communication, allowing to infer the approximate answer to a question based on knowledge that is inexact, incomplete or not completely trustworthy. The use of fuzzy sets, which are naturally inclined to deal with the domain's linguistic knowledge, can produce easier to interpret solutions [23], which allows you to create specialist systems by using linguistic variables. Fuzziness is found precisely in information of this nature [19].

In fuzzy logic, a function must be generalized to be able to assume values in a given interval and the assumed value indicates the pertinence of an element in a particular set. In this way, the pertinence degree function μA of a fuzzy set A is in the form: μA: $X \rightarrow [0,1]$. In other words, fuzzy set A is characterized by the pertinence function $\mu A(x)$, which assigns a real number in the interval $[0,1]$ to each element in the set X. In this way, the value of $\mu A(x)$ represents the degree of pertinence of element $x \in X$ in set A [14]

There are important works that use fuzzy logic in data mining. In [21], this technique is used in decision-making systems and marketing systems. Fuzzy logic has also been used to analyze consumer behavior [16, 29]. [17] shows that fuzzy logic is the most adequate mathematical model for the treatment of data in a study that tried to reproduce consumer behavior in choosing brands in a virtual supermarket, when compared to conventional methods, such as boolean logic models relying on determinism and probability.

The problem of ambiguity in text processing can be tackled with the use of fuzzy logic, as its purpose is to deal with imprecise situations, providing improved results by way of the pertinence calculation of an element to a set. By using this technique, it is possible to define just how important and relevant a term is (or not) to any given category. There are a number of fuzzy functions that can be used to fulfill this end. The simplest fuzzy function is called set theoretic inclusion. [4] assesses the presence of words in two documents, which are compared to one another. If the term appears in both documents, the value of 1 is added to the counter; if not, 0 is added. At the end, the degree of similarity is a fuzzy value between 0 and 1, calculated by mean, *i.e.*, the total value of the term counter divided by the total number of words that appear in both documents. This fuzzy value represents the degree in which one element is included in the other or the degree of equality between them. However, this function presents a problem, since it only weights the importance of a word appearing in both documents. The fact that a given word is more or less important in one document than in another, as it appears in different frequencies, is not taken into account. This problem can be partly resolved by using another function, which calculates the mean using fuzzy operators, which are similar to the above function, but assigning weights to the terms [25]. Thus, the fact that the terms appear with different levels of importance is taken into account. In this case, the weights of the terms may be based on the relative frequency or any other discriminating value. Both these functions are found in the literature and were used separately. However, in this work, they will be used together. More details can be found in Section 4.

The use of fuzzy logic in this work is focused on categorizing elements, not only in terms of pertinence or non-pertinence, as in the case of classical theory, but also in terms of varying degrees of pertinence. Hence, the fuzzy approach is used to cate-

gorize objects in accordance to a measure of similarity between them and the center of a conceptual space, whereby the closer to the center the object is, the more similar it is; the further away from the center, the less similar. Each text is represented by a set of characteristics that best define it, and fuzzy similarity is then used to define how similar two representative vectors are. Based on a set of characteristics of a text, composed here by the attributed relevance of the terms in relation to the text, the fuzzy approach is founded upon the notion of similarity of text and a category. The results that are supplied are partial classifications, where each category is assigned a degree of pertinence or relevance in relation to the analyzed text. To verify the similarity between a text and a category, all the terms that make up the set of characteristics of the text are compared to the terms that make up the set of characteristics of the category. A term is considered similar when it is found in the index of the category as well as in the index of the text. The degrees of equality of the terms are then used to determine the degree of similarity between the text index and the category index. In this way, the text is classified under the category in which it obtains the highest degree of similarity. Section 4 will explain this proposal, as well as the functions mentioned above, in more detail.

3 Agglomerative hierarchical methods

The clusterization process can be defined as a process that acccept as input continuous regions of a space that has a large number of points and divides this regions into regions with smaller amount of points, called clusters. These clusters have the following properties: density, variance, dimension, form and separation. Based on these properties, different types of conglomerates emerge, which may be hyperspherical, curvilinear, elongated or they may have structures that are more differentiated [1, 7, 3]. According to their configuration, the clusters can be classified into the following categories: hierarchical agglomerative, hierarchical divisive, iterative partitioning, density search, factor analytic, clumping and graph-theoretic. When applied to a data set, these algorithms generate different results [1, 6, 3].

In hierarchical methods, the data are partitioned successively, producing a hierarchical representation of the clusters. This type of representation makes it easier to visualize the clusters at each stage, as well as facilitates the perception of the degree of similarity between them. Another interesting characteristic is that hierarchical methods does not require a definition of the number of clusters. The main advantage of this method [3] is that different similarity measures can be used, which augments the applicability of these methods to any type of attribute (numeric or categorical). The main disadvantages are the stop criterion and the non-refining of the results as the hierarchy is being constructed. With regards to the stop criterion, this can be defined when one reaches a given number of clusters or when some type of stop condition takes place. This criterion requires a distance matrix between the clusters, known as a similarity matrix [15]. This similarity matrix characterizes ano-

ther problem in the hierarchical methods because it grows exponentially in the face of its database [35].

To calculate the distance in the similarity matrix, many methods can be used [2]. The most important ones are: Simple Connection (the distance between two more similar clusters); Complete Connection (the distance between two less similar clusters); Centroids (the distance between two clusters is obtained by their centroids); Connection Mean (the mean of the distance s between elements of each cluster); Connection group Mean (the distance of two clusters is obtained by the mean of the union of two related clusters) and Ward (finding partitions that minimize the loss associated to each cluster).

In this work, two approaches in hierarchical methods have been considered: agglomerative (Bottom-up) and divisive (Top-down) ones [1, 7, 3]. In the agglomerative hierarchical approach, the data are initially distributed in such a way that each example represents a cluster and, then, these clusters are recursively clustered taking into consideration some measure of similarity until all of the examples belong to one single cluster. Hence, in the beginning, the clusters exist in reduced numbers with a high degree of similarity between their elements, but throughout the process, these groups start to increase and their elements become dissimilar [31]. In Algorithm 1, the steps that are conducted in this approach are described. In this way, Figure 1 can be interpreted as initially containing five clusters [A, B, C, D, E]. At the end of all the steps, a cluster called G1 is formed, wherein clusters [A,B] can be found and the similarity of the G1 cluster is measured by Distance D1. The cluster G2 is formed by the clusters [D,E], in which case the measure of similarity for G2 is equal to D2. In the next step, cluster G3 is formed by the cluster [C] and by the cluster G2 and the similarity distance of G2 to G3 is the distance D3. The next step is to create the cluster G4, formed by clusters G1 and G3, and the similarity distance is D4. An agglomerative hierarchical algorithm can basically be described in the following way:

1 - Agglomerative Algorithm:
1. Look for the pair of clusters with the largest degree of similarity.
2. Create a new cluster that groups the selected pair in step 1.
3. Decrease by 1 the number of remaining clusters.
4. Return to step1 until only one cluster is left.

The divisive hierarchical method, on the other hand, is the least common among the hierarchical methods, as it is inefficient and has high computational costs [1, 6, 3]. In the divisive hierarchical approach (Figure 2), the process is initiated with only one cluster, which contains all the data, and continues to recursively divide according to a given metric that reaches a given stop criterion, usually the number of clusters that are wanted [17]. Figure 2 can be interpreted as, in the beginning, everyone is in the cluster [G4] making up one single cluster. This cluster is divided into two clusters [G1 and G3] and the similarity measure is represented by D1. In the next step, one can see that the cluster G3 is divided into [C and G2] and the measure of similarity between these clusters is D2. At this point there are already

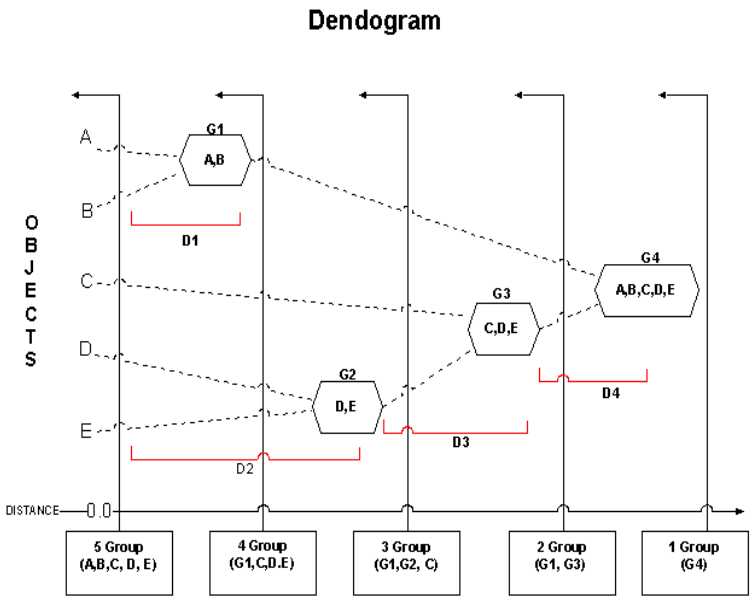

Fig. 1 Dendogram of the Agglomerative Hierarchical Method.

three clusters [G1, G2 and C]. The cluster [G2] is divided into [D and E] and the similarity between these clusters is the distance D3. In this case, four clusters remain [G1, C, D, E]. The next step is to divide cluster G1, creating clusters [A and B] and the measure of similarity is expressed as the distance D4. At this point, we are left with the five clusters [A, B, C, D and E]. The steps to this approach are described in Algorithm 2.

2 - Divisive Algorithm:
1. One single cluster containing all elements is constructed;
2. The similarity matrix is calculated between all pairs in the cluster;
3. A new cluster is created dividing the pairs with the lowest degree of similarity;
4. Return to Step 1 until each cluster contains a single element, or the desired number of clusters is achieved.

The most important algorithms pertaining to the agglomerative hierarchical method, according to [20], are: Cliques, Stars, Connected Components and Strings.

The biggest problem in Natural Language Processing methods is its complexity. They involve the analysis of a series of issues such as text coherence and cohesion, which could be related to cultural, social, situational and political issues and/or they could be directly related to the author and the moment in which the text was written [11]. On algorithmic view, texts are analyzed in clusters for the purpose of information recovery or knowledge discovery. It is necessary that the groups constituted

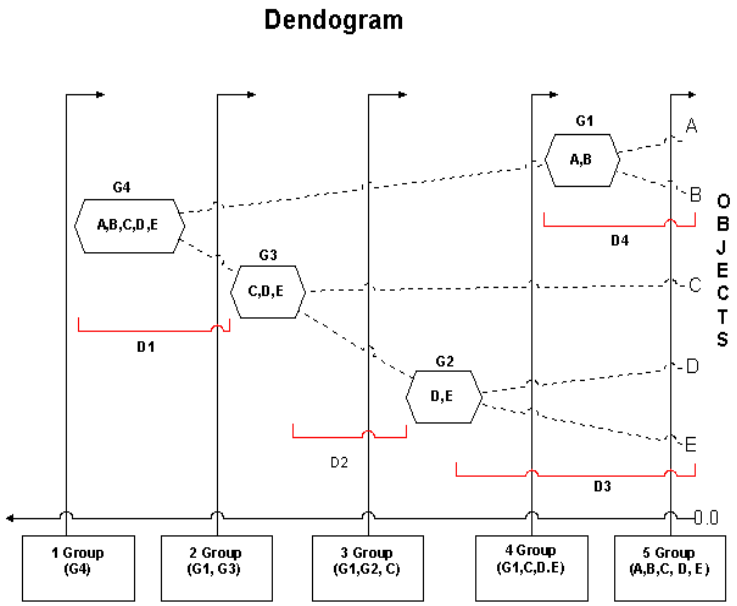

Fig. 2 Dendogram of the Divisive Hierarchical Method.

by the texts (objects) have a certain cohesion amongs them. The clusters with very different objects would not be admissible due to the lack of cohesion of their texts. The problem is that some of the algorithms, such as Connected Components and Strings, are not as restrictive as expected [37], because they allow objects with a small degree of similarity to be clustered simply because they have a strong relationship with one single object in the group, but not with all the objects found in the clusters. Hence, in this work, we choose using Cliques and Stars algorithms due to their ability to construct more cohesive clusters; that is, texts that are more coherent among themselves. In what follows, we describe in details both algorithms.

3.1 Stars Algorithm

The Stars algorithm [20] has this name precisely because the conglomerates that are formed have a shape that is similar to a star; that is, one central element with a variety of other elements connected to it, creating the tips of a star. In this case, the central element is the one that has a relationship to all the other elements of the star, which are interconnected. The elements at the tips are not necessarily related one to the other, which is precisely one of the algorithm's biggest shortcomings, seeing as the elements may not be similar. To minimize this problem of the lack of similarity between the elements located on the tips of the star, a similarity threshold

must be established. Hence, the solution for the elements on opposing tips of the star not to be too dissimilar or distant consists of selecting a larger degree of similarity, seeing as the closer they are to the center, the more similar the elements will be amongst themselves, giving the group more coherence. The star algorithm is shown in Figure 3. Algorithm 3 describes the steps in the Star algorithm.

3 - Star Algorithm:
1. Select 1 (one) element and place all similar elements in the same cluster;
2. Elements that are not yet allocated/classified are placed as a cluster seed (repeat step 1 for 1 element that is not yet allocated).

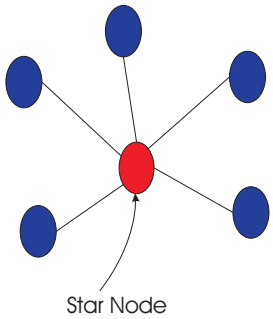

Fig. 3 Graphic representation of the Star Algorithm.

3.2 Cliques Algorithm

The Cliques algorithm [20], whose graph when formed is illustrated in Figure 4, is similar to the starts algorithm, however, the elements are only added to a cluster IF their degree of similarity is greater than the threshold for all the elements already present in the conglomerate, not only in relation to the central element. In this case, the conglomerates tend to be more cohesive and to have a higher quality, seeing as the elements are more similar or closer to one another. Algorithm 4 describes the steps of the Cliques algorithm.

4 An Approach to Text Categorization — A Proposal

This section proposes an approach to text categorization. This approach is divided into four steps, as illustrated in Figure 5.

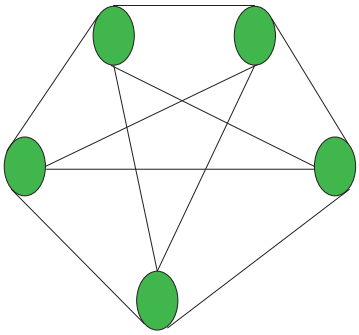

Fig. 4 Graphic representation of the Cliques Algorithm.

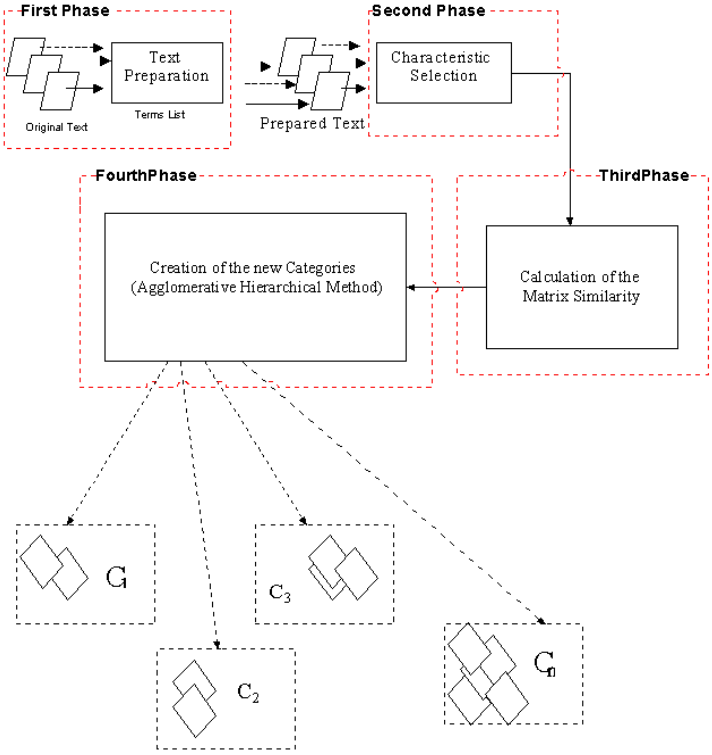

Fig. 5 Use of Stars and Cliques algorithms in the Agglomerative Hierarchical Method.

4 - Cliques Algorithm:
1. Select next Object and add it to a new cluster;
2. Look for a similar object;
3. If this object is similar to all of the objects in the cluster, add it;
4. Stop criterion: while there is at least one object not allocated, come back to Step 2;
5. Return to step 1.

In the first step, a pre-text-processing stage is conducted, in which the texts are prepared for the second step. In this step (Step 1), a technique, called case folding, which consists of transforming all words into small case letters, is used. After, the stopwords [26][1] are removed. The purpose of this step is to make the text more concise and the category index more succinct. The removal of stopwords as well as the case folding technique in Text Mining were proposed by [36].

In the second step, term characteristics in the text are selected by way of relative frequency. The latter defines the importance of a given term according to the frequency in which the term appears in the text. The more a term appears in a text, the more important it is in defining it. It is due to this definition of relative frequency that the removal of the stopwords is so important in the pre-processing step. The relative frequency is calculated by way of Equation 1 [30]. This formula normalizes the result of absolute frequency of the terms by preventing small documents to be represented by small vectors and, conversely, large documents be presented by large vectors. After this normalization, all the documents will be represented by vectors of the same size.

$$F_{rel}X = \frac{F_{abs}X}{N} \tag{1}$$

Where:

- $F_{rel}X =$ relative frequency of X;
- $\frac{F_{abs}X}{=}$ absolute frequency of X, that is, the amount of times in which X appears in the document;
- $N =$total number of terms in the text.

Since a vectorial-space is considered, where each term is represented by one dimension, there are as many dimensions as there are different words. Even when we eliminate the stopwords, one of the biggest problems encountered in TM is dealing with the very large dimension spaces. In this way, one of the important problems handled in the second step of this approach is the reduction of dimensionality. In order to do this in this work, we adopted a minimum importance value, a threshold or similarity threshold [37], in which the words (characteristics) with an importance (frequency) below the given value (threshold) are simply ignored. This technique is important given the high dimensionality of the space of characteristics, that is,

[1] Stopwords are closed classes of words that do not carry meaning, such as articles, pronouns, interjections and prepositions.

the large volume of words that compose a document must be treated. Therefore, in order to attain a better categorization, it is necessary to reduce the space.

The third step aims to identify the similarity between the terms (the characteristics selected in the second step). To this end, a measure of fuzzy similarity was used: a measure, called set theoretic inclusion [4], which evaluates the presence of words in the two elements (texts) that are being compared. If the term is present in both elements, a value of one (1) is added to the counter; if it isn't, zero (0) is added. At the end, the degree of similarity is a fuzzy value between 0 and 1 calculated by the mean; that is, the total value of the common term counter divided by the total number of words in both documents (without counting repeated terms). After calculated the fuzzy similarity, a matrix is generated that indicates the similarity values between every text present in the text database. In the main diagonal of the similarity matrix, the value is always 1, as the degree of similarity of a text when compared to itself is always 1. Based on this matrix, clustering algorithms are used to identify the text clusters, which specify some type of relationship rule.

The fourth and final step of the proposed approach consists of using the agglomerative hierarchical method, whose main advantage upon the other clustering methods is the non-definition of a prior number of clusters. Analyzing the constructed dendograms, it is possible to work out the appropriate number of clusters. We used the Cliques and Star algorithms, as these algorithms are capable of constructing more cohesive clusters, as seen in Section 3.

In the next section, we will describe the experiments conducted with the approach proposed in this work, which are compared to the categorizer proposed by [37], called Eurekha.

5 Experiments

For the experiment with the Categorizer proposed in this work and Wives's Eurekha categorizer [37] the following Corpus were used: TeMario [26], Reuters-21578, Distribution 1.0 and Really Simple Syndication (RSS)[2].

Figure 6 illustrates the composition of the TeMário Corpus. This corpus is composed of two main sections: summaries and source texts. The source-text section is subdivided into source texts with titles, source texts without titles and source texts with subdivided titles. These were separated into two categories: Folha de São Paulo and Jornal do Brasil, two newspapers with major circulation in Brazil. In this summaries section, there are: manual summaries, ideal summaries and marked summaries. Marked summaries contain sections which an automatic summarizer should select from the original text. To conduct the experiments, we used source texts with the subdivided titles, illustrated in Figure 6 in the box with dotted lines. Regarding to the sub-division, the texts of each newspaper with subdivided into 5 categories:

[2] Corpus extracted from Terra Networks Brasil S/A.

Special, World, Opinion, Politics and International. Each of these categories possesses a total of 20 texts.

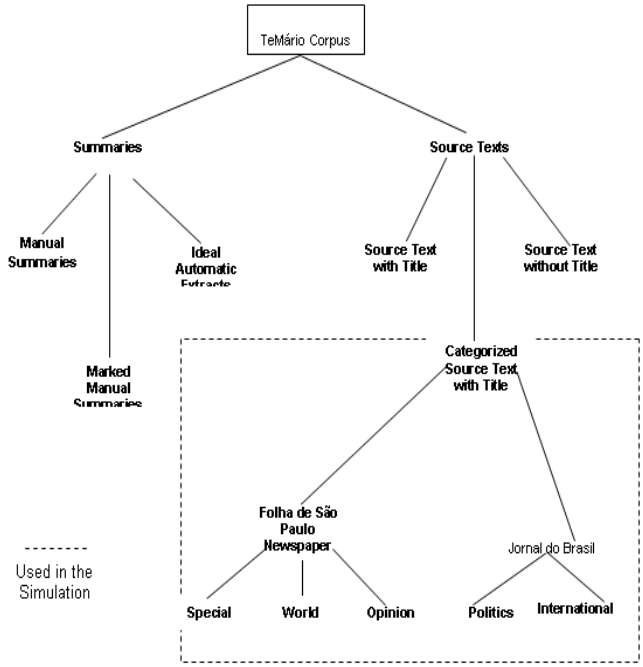

Fig. 6 Division of categories in the TeMário Corpus.

The Reuters Corpus is made up of 100 texts in English, all in the field of economics[3]. The Distribution 1.0 corpus has 22 files. The RSS_Terra corpus is also made up of 100 texts, in Portuguese, classified in 7 categories: Brazil (22 texts), Cities (16 texts), Education (1 text), Police (36 texts), Politics (13 texts), Health (8 texts) and Traffic (4 texts)[4].

5.1 Hypothesis

The null hypothesis of this work consists in the statement that the Categorizer is equal to Eurekha when it comes to text distribution in the categories. This hypothesis is true for both the use of the Cliques algorithm as well as for the Stars algo-

[3] The complete collection has 1,578 texts, however, these files were not available for use in their totality. Hence, we used only the 100 texts that are available online.

[4] These files, which come from the most diverse RSS channels of Terra Networks Brasil S/A, were collected daily during the period comprising February to March 2008.

rithm and for each of the corpuses that were simulated. It relates the variance in the number of texts of each category constructed by the categorizers. In other words, if a categorizer found 3 categories, each with 3, 5 and 7 texts, the variance of this sample of the population is 4. Formally, this null hypothesis can be represented by Equation 2.

$$H_0 : \sigma\ Categorizer = \sigma\ Eurekha \tag{2}$$

Where:

- $H_0 =$ null hypothesis
- $\sigma\ Eurekha$ variance of the Eurekha sample,
- $\sigma\ Categorizer$ variance of the Categorizer sample.

If the null hypothesis is considered false, some other statement must be true. Hence, this work proposes an alternative hypothesis H1 that represents the opposite of the null hypothesis H0. The alternative hypothesis is formally represented by Equation 3.

$$H_1 : \sigma\ Categorizer \neq \sigma\ Eurekha \tag{3}$$

The methodology to test the hypothesis that was adopted in this work considers the populations, which were obtained in the generated categorizer simulations, independent and with the same variability. Hence, the F-test was chosen, where the populations were assumed to be normally distributed and the ration of the variance of the samples follow a distribution known as F [32].

5.2 Decision Rule for the F-test

The critical values of the F distribution depend on two sets of degrees of freedom. The degrees of freedom of the numerator of the fraction pertain to the first sample (Eurekha), and the degrees of freedom in the denominator pertain to the second sample (Categorizer).

The null hypothesis is rejected if the statistics of the F-test are calculated as being greater than the critical value of the upper tail, F_S, based on the distribution of F with $n_1 - 1$ degrees of freedom in the numerator, from Sample 1, and $n_2 - 1$ degrees of freedom in the denominator, from Sample 2.

The null hypothesis is also rejected if the statistics of the F-test are positioned below the critical value of the lower tail, F_I, of the distribution of F, with $n_1 - 1$ and $n_2 - 1$ degrees of freedom in the numerator and in the denominator, respectively.

Therefore, the decision rule is:

Reject H_0 if $F > F_S$ or $F < F_I$;

If not, do not reject H_0.

Figure 7 shows the areas of rejection and non-rejection, keeping in mind that this is a two-tailed test and the area of rejection is shared between the lower and

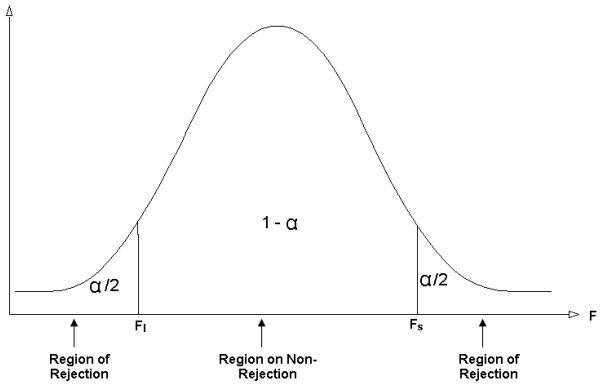

Fig. 7 Regions of rejection and non-rejection for the two-tailed F test.

upper tails of the F distribution. Since, in this work, we have adopted the level of significance of 5% with a value of $\alpha = 0,05$, then the region of rejection will contain $0,025$ of the distribution, in other words, $\frac{\alpha}{2}$.

5.3 Testing the Null Hypothesis

The procedure for testing the hypothesis of equality of the two variances is based on the following result: Let $x_{11}, x_{12}, ..., x_{1n}$ be a random sample of a normal population with a mean of $\mu 1$ and variance of σ_1^2, and let $x_{21}, x_{22}, ..., x_{2n}$ be a random sample of a second normal population with a mean of $\mu 2$ and variance of σ_2^2. Assume that both populations are independent. Let S_1^2 and S_2^2 be the variances of the samples. The ratio

$$F = \frac{S_1^2}{S_2^2} \tag{4}$$

have a distribution F, with $n_1 - 1$ degrees of freedom in the numerator and $n_2 - 1$ degrees of freedom in the denominator. This result is based on the fact that $\frac{(n_1-1)S_1^2}{\sigma_1^2}$ is a random variable with $n_1 - 1$ degrees of freedom, that $\frac{(n_2-1)S_2^2}{\sigma_2^2}$ is a random variable with $n_2 - 1$ degrees of freedom and that both populations are independent.

The idea of the null hypothesis in this work $H_0 : \sigma\ Categorizer = \sigma\ Eurekha$ where the ratio $F = \frac{S_1^2}{S_2^2}$ with a distribution $F = \frac{n^1-1}{n^2-2}$. Formally, this can be represented:

- $S_1^2 =$ variance of the sample of n_1 elements;

- $S_2^2 =$ variance of the sample of n_2 elements;
- Degree of freedom is given by:
  $F_S = n_1 - 1 =$ degree of freedom in the numerator;
  $F_I = n_2 - 1 =$ degree of freedom in the denominator;

The following formulas are obtained for the calculations:

$$F = F_{(\frac{\alpha}{2}, n_1-1, n_2-1)} = Critical\ limit\ of\ the\ uppertail \tag{5}$$

$$F = F_{(1-\alpha, n_1-1, n_2-1)} = Critical\ limit\ of\ the\ lowertail \tag{6}$$

Each corpus has a population of 100 texts. Their samples correspond to the distributions of each text in numbers of categories created by Eurekha and by the Categorizer, using the algorithm Star and Cliques on each of the simulated corpora. As an example, consider the Reuters corpus and the Cliques algorithm, for which the Eurekha categorizer obtained 15 categories, while the Categorizer obtained 38. In this way, there are (15-1) degrees of freedom for the Eurekha categorizer and (38-1) degrees of freedom for the Categorizer.

The F_S of each corpus, the critical value of the upper tail of the F distribution is obtained by Equation 5. In [28], one is able to locate the Table showing the distribution values of F.

In F_I, the critical value of the lower tail of the F distribution, with $n_1 - 1$ degrees of freedom, from Sample 1 in the numerator and $n_2 - 1$ degrees of freedom from Sample 2 in the denominator, is calculated by taking the reciprocal of F_S*, a critical value of the upper tail of the F distribution, with "inverted" degrees of freedom, that is, $n_2 - 1$ degrees of freedom in the numerator and $n_1 - 1$ degrees of freedom in the denominator. This relationship is shown in Equation 6.

Let us return to the example in order to show how the F test works. Recalling that the degrees of freedom are equal to 37 and 14, respectively, to obtain the critical value of $0,025$ from the lower tail, you need to obtain the critical value of the lower tail, which, in this case, equals $2,27$, with 37 degrees of freedom in the numerator and 14 degrees of freedom in the denominator. Hence, the value of $F_I = \frac{1}{2,43} = 0,412$. Using the decision rule, we have:

Reject H_0 if $F > F_S = 2,27$ or $F < F_I = 0,412$
If not, do not reject H_0.

In Equation 4, the ratio of the proportion of the two samples is calculated. Applied to the example of the Reuters corpus, we have: $F = \frac{10.809521}{3,20056} = 3.37738$. Therefore, in the example of the Reuters corpus we have $F_I = 0,412 < F = 3,37738$. As $F = 3,37738 > F_S = 2,27$, H_0 is rejected, a significant difference between the variability of Eurekha and of the Categorizer does exist in the text distribution for each of the categories created in the Reuters corpus simulation.

Tables 1 and 2 display the results of the F-test using the Reuters, TeMário and RSS_Terra corpuses and the algorithms Star and Cliques. A 95% trust interval was established for this two-tailed test.

Table 1 Results of the F-test applied to the simulation corpuses with the Star algorithm with a degree of significance of 5%.

Star			
Corpus	TeMário	Reuters	RSS_Terra
F_I	$0,42$	$0,05$	$0,5$
F_S	$2,34$	$19,44$	$2,00$
F	$3,17$	$102,08$	$1,86$
H_0	$Reject(F > F_S)$	$Reject(F > F_S)$	$Accept$

Table 2 Results of the F-test applied to the simulation corpuses with the Cliques algorithm with a degree of significance of 5%.

Cliques			
Corpus	TeMário	Reuters	RSS_Terra
F_I	$0,55$	$0,41$	$0,57$
F_S	$1,80$	$2,27$	$1,75$
F	$0,44$	$3,38$	$11,66$
H_0	$Reject(F < F_I)$	$Reject(F > F_S)$	$Reject(F > F_S)$

After analyzing the results shown in Tables 1 and 2, you can see that the null hypothesis was only accepted in the RSS_Terra corpus using the Star algorithm. When the null hypothesis is accepted according to the F-test, a t-test is indicated for the difference between the two arithmetic means with the equal variances. For the t-test, assuming that there are two populations with unknown means of $\mu 1$ and $\mu 2$, we have:

$$H_0\ (null\ hypothesis)\ : \mu 1 = \mu 2$$
$$H_1\ (alternative\ hypothesis)\ \mu 1 \neq \mu 2$$

The t-test is formally described by:

$$t = \frac{(\overline{X_1} - \overline{X_2})}{S_a \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{7}$$

where

$n_1 =$ size of sample 1;
$n_2 =$ size of sample 2;
$\overline{X_1} =$ mean of sample 1;
$\overline{X_2} =$ mean of sample 2;
$S_1^2 =$ variance of sample 1;

$S_2^2 =$ variance of sample 2;
$S_a =$ clustered variance is calculated by:

and

$$S_a \frac{(n_1-1)S_1^2+(n_2-1)S_2^2}{n_1+n_2-2} \tag{8}$$

Cluster variance is given this name because the statistics of the test require that both variances of the sample, $S_1^2 = S_2^2$, be clustered or combined for the purpose of obtaining S_a^2, the best estimate of a variance that is common to both samples, under the premise that both variances of the two samples are equal.

The t-test statistics for cluster variance follows a t distribution with n_1+n_2-2 degrees of freedom. In this way, the criteria for the rejection of the null hypothesis can be formalized as follows:

$$\text{Reject } H0,\ if\ t\ >\ t_{n_1+n_2-2}\ or\ t < -t_{n_1+n_2-2}$$

Table 3 shows the result after confirming the null hypothesis for the variances ($H0:\ \sigma_1^2\ =\ \sigma_2^2$) in the simulation of the RSS_Terra corpus, where we applied a t-test with a significance level of 5% to test the difference between the means ($H0:\mu 1=\mu 2$).

Table 3 Results of the t-test applied to the RSS_Terra corpus of the simulation with the Star algorithm with a degree of significance of 5%, which obtained equal variance.

Corpus-Star	RSS_Terra
S_a^2	$15,22$
t_I	$-2,00$
t_S	$2,00$
t	$1,40$
H_0	*Accept*

With the result obtained in the t-test, it became clear that the means of both populations were effectively equal. Hence, the probability of detecting a difference with this dimension or greater, between the two arithmetic means of the samples, corresponds to 0.19806804. Since the critical value is greater than $\alpha = 0,05$, there isn't sufficient evidence to accept the null hypothesis.

5.4 Qualitative analysis of the constructed categories

In Figure 8, the graph shows the number of categories created by Eurekha and by the Categorizer, using each of the three simulation corpuses with the use of the Star and Cliques algorithms.

Figure 8 shows that the Categorizer obtained in each of the simulated corpuses a greater number of categories in comparison to Eurekha.

The Categorizer obtained in the Reuters and RSS_Terra corpuses a greater number of categories as compared to Eurekha. In this way, Eurekha only had a higher number of categories in the TeMário corpus using the Cliques algorithm, which accounts for 16.66% of all the simulations.

After analyzing the results in Figure 8, some important observations can be made regarding the amount of categories obtained in the Categorizer and in Eurekha:

The first aspect refers to the methodology adopted in this work, which opted for clusterization using the agglomerative hierarchical method. This technique is important in this work precisely due to the fact that it does not define the initial number of clusters since, in the context of Text Mining, the domain specialist would have to define how many categories there would be to later start categorizing. This process creates a certain degree of autonomy as there is no need for human intervention in the act of defining the number of categories, as these are automatically generated by way of the agglomerative hierarchy.

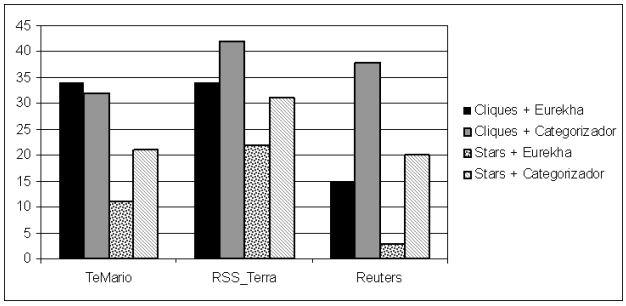

Fig. 8 Number of categories created by Eurekha and by Categorizer using the Star and Cliques Algorithm.

The second aspect worth noting refers to the other part of the methodology proposed in this work, where a minimum value of importance, a threshold (here, we used $0,05$) or similarity threshold [37] was employed in which the words (characteristics) with an importance (frequency) below a given value are simply ignored. Along with the threshold, the use of fuzzy similarity, that is, the measure of set theoretic inclusion, determines the number of categories and sub-categories that will exist throughout the process and also determines the similarity distance between them.

The high number of categories represents the refinement of the texts. Texts categorized using our approach are added into one category only if its similarity rate is bigger than the boundary to all the texts present in the category and not only related to the main category. This factor is significant to indicate the high degree of similarity among the clustered texts and also shows that the greater the distance between the categories (depth level in the hierarchy tree), the greater will be the dissimilarity between them, thereby determining the higher degree of similarity between the texts

grouped into each category. Hence, this proves the justification that the higher the number of categories, the greater the refinement among the categorized texts.

5.4.1 Details of the results from the Categorizer with the Star Algorithm using the Reuters Corpus

Using the Stars algorithm, the Categorizer created 20 categories. Category C_1 was the one that obtained the highest number of texts - 18 in total - and their topics were the economy. Categories C_{12}, C_{17}, C_{18}, C_{19} and C_{20} were the categories with only one allocated text. It was observed that for the texts that were clustered in pairs, as in the case of category C_{11}, there was no coherence with regards to their topics. Now with only two texts there is category C_{11} which appears to have texts that are closely related and handle the same topic.

There were 15 texts in category C_2, all of which related to political economy in general. In C_3 there was a total of 10 texts whose main topic involves types of investments. The Categorizer clustered into category C_5 11 texts on investments focused exclusively on companies looking for patents and new products. One can see that some of the files speak a lot of pharmaceutical labs. In category C_{10}there were 8 texts that talk generally about the economy in a variety of different countries. In category C_{14} there were 5 texts that deal essentially with production. Categories C_8, C_9 and C_{13} each had a total of 4 texts. The ones in C_3 covered mining, agriculture and the market; the Japanese economy was the topic of the texts in C_9 and, in C_{13} , no relationship was found between the texts.

With 3 texts grouped into each category, we have categories C_4, C_6, C_7, C_{15} and C_{16} although in categories C_7, C_{15} and C_{16} there was no relationship between the texts. However, the texts in category C_4 cover the world economy and cite Argentina, Tanzania and Africa. Category C_6 handles joint ventures with Japan.

5.4.2 Details of the results from Eurekha with the Star Algorithm using the Reuters Corpus

In this simulation, the Eurekha categorizer created 3 categories, wherein C_1 obtained a total of 86 texts, C_2 clustered 12 texts and C_3 had 2 texts. By the analysis, it is not possible to establish a relationship between the texts clusters in C_1, C_2 and C_3, as there is no apparent relationship between the texts. We were unable to establish coherence in the texts within the referred clusters.

5.4.3 Graphically Comparing the results of the text distribution in the Categorizer and in Eurekha with the Cliques Algorithm using the Reuters Corpus

As we can see in Figure 9, the Eurekha categorizer obtained a total of 15 categories, while the Categorizer had 38 categories. If we observe the text distribution in the Categorizer, we can see that there was no category with more than eight allocated texts, whereas using Eurekha, 13 texts were allocated to the C_{12} category.

We also can see that in the Categorizer there were 10 categories that only had one single allocated text, whereas in using Eurekha there were no categories with only one single text. However, the number of categories created by the Categorizer was more than double the amount of categories created by Eurekha.

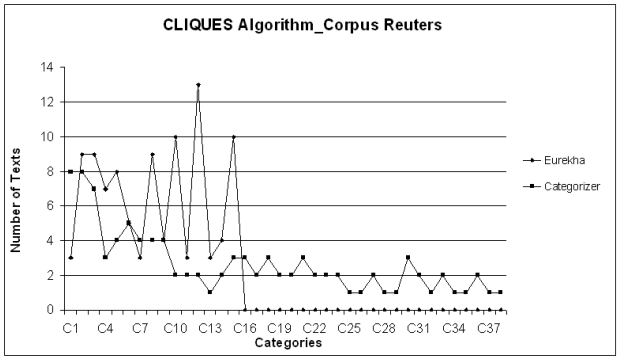

Fig. 9 Number of categories created in the CLIQUES algorithm with the simulation of the Reuters corpus in the Categorizer and in Eurekha.

5.4.4 Comparing the results of the text distribution of the Categorizer and of Eurekha with the Star Algorithm using the RSS_Terra Corpus

According to Table 4 the Categorizer obtained a greater number of categories in comparison to Eurekha. In Table 6 you can see how the texts were distributed in the categories created by each of the categorizers when using the Star algorithm.

Table 4 Total Number of categories created in the categorizers Eurekha and Categorizer in the RSS_Terra corpus.

Amount of Categories created	
Eurekha	Categorizer
22	31

Table 5 shows that the Categorizer had a distribution of 16 texts in one given category, while Eurekha obtained in a given category of cluster of 20 texts. In Eurekha there was no cluster in categories with 1, 8, 10 and 11 texts, whereas in the Categorizer this did not occur with the amounts of texts 5, 7, 8 and 10. The Eurekha categorizer did not have any category with only 1 text, but this occured in 12 categories in the Categorizer.

Table 5 Text Distribution in the categorizers created by Eurekha and Categorizer using the RSS_Terra corpus.

Amount of Categories created	
Eurekha	Categorizer
34	42

5.4.5 Comparing the results of the text distribution of the Categorizer and of Eurekha with the Cliques Algorithm using the RSS_Terra Corpus

According to Table 6, the Categorizer obtained a greater number of categories in comparison to Eurekha. Table 7 shows how the texts were distributed in the categories created by each of the categorizers when using the Cliques algorithm.

Table 6 Total Number of categories created in the categorizers Eurekha and Categorizer in the RSS_Terra corpus.

Text Distribution in the Categories			
Eurekha		Categorizer	
Amount of Text	Amount of Categories	Amount of Text	Amount of Categories
1	0	1	12
2	5	2	6
3	8	3	4
4	2	4	5
5	3	5	0
6	0	6	1
7	2	7	0
8	0	8	0
9	1	9	1
10	0	10	0
11	0	11	1
20	1	16	1

Table 7 shows that the Eurekha categorizer clustered 20 texts in only one single category, whereas the Categorizer placed in one given category 6 texts. None of the categories in Eurekha received only one text, whereas in the Categorizer there were 11 categories with only one single text. In Eurekha texts in pairs occurred in 24 categories, whereas in the Categorizer they were found in 14 categories. Eurekha

did not create a single category in which the amount of texts was 1, 5 and 6, whereas in the Categorizer, each category that was created had at least one allocated text.

Table 7 Text Distribution in the categorizers created by Eurekha and Categorizer using RSS_Terra corpus.

Distribution in the Categories			
Eurekha		Categorizer	
Amount of Text	Amount of Categories	Amount of Text	Amount of Categories
1	0	1	11
2	24	2	14
3	8	3	10
4	2	4	5
5	0	5	1
6	0	6	1
20	1	–	–

5.4.6 Comparing the results of the Categorizer in the distribution of texts in each category using the TeMário corpus in relation to the Star and Cliques algorithm

The use of the Cliques algorithm by the Categorizer generated 32 categories, as shown in Table 8. Category C_4 and C_{10} were the ones that obtained the highest number of texts using the Categorizer: 9 each. With the Star algorithm, 23 categories were created. Category C_1 obtained a total of 21 texts. Categories C_{13}, C_{25}, C_{27} and C_{31} were categorized with only a single text. The remaining categories had their texts clustered in intervals ranging from 2 to 8, as indicated in Table 9. All of them are characterized by coherence in their subjects. In contrast, for the Star algorithm, categories C_5, C_{10}, C_{11}, C_{17}, C_{18}, C_{19}, C_{20} and C_{23} were clustered with only one text, while the other categories had texts clustered in intervals between 2 and 21.

Another fact that must be observed in the Cliques algorithm is the lower number of categories created with only one text (4 in total) in contrast to the Stars algorithm (which created 8).

No incoherence was observed in the Cliques algorithm in the case of categories with only two texts, as seen with the Stars algorithm.

As seen in Figure 10, the Cliques algorithm, in comparison to the Stars algorithm, didn't have any category with a cluster of over 10 texts. This is due to the fact that, in this algorithm, elements are only added to a category if their degree of similarity exceeds the threshold for all elements already present in the category and not only with regards to the central element.

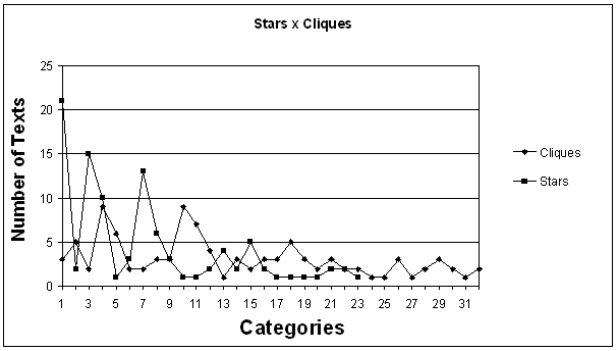

Fig. 10 Comparison between the stars and Cliques algorithm using the Categorizer on TeMário corpus.

5.4.7 Comparing the results of Eurekha in the distribution of texts in each category using the TeMário corpus in relation to the Star and Cliques algorithm

Eurekha generated 33 categories using the Cliques algorithm and 10 categories with the Star algorithm. By analyzing the behavior of the Cliques algorithm in Table 9, one can see that there is a certain uniformity in the text distribution in each category, which is normal for this algorithm. The maximum was 6 (six) texts allocated per category and there was no case of a category that was allocated only a single text. In contrast, with the Star algorithm there was a very high concentration of texts in category C_1 (with 32 texts) and in C_2, C_3 and C_6 (with 13, 18 and 10 texts, respectively).

Another fact that must be observed in the Cliques algorithm is the very high number of texts allocated in pairs, which occurred in 16 categories. Furthermore, the greatest text allocation took place in category C_{12}, with 6 texts, wherein five were in the international category and one in the world category.

In the Star algorithm, the category that received the lowest number of allocated texts was C_8 with only two texts. The most noteworthy characteristic of this algorithm is the very high concentration of texts in the initial categories, as can be observed in Table 9.

As seen in Figure 11, the Cliques algorithm, in comparison to the Stars algorithm, didn't have any category with a cluster of over 6 texts. This is due to the fact that, in this algorithm, elements are only added to a category if their degree of similarity exceeds the threshold for all elements already present in the category and not only with regards to the central element.

Table 8 Texts clustered by the Categorizer using the Star (S) and Cliques (C) algorithm in the TeMário Corpus. *Folha de São Paulo* is abbreviated as FSP and *Jornal do Brasil* is abbreviated as JB.

Source Text with Origin and Title													
Categories		**FSP**						**JB**				Total	
Created		**Opinion**		**World**		**Special**		**Intern.**		**Politics**		Categories	
S	C	S	C	S	C	S	C	S	C	S	C	S	C
C_1	C_1	13	3	2	-	5	-	-	-	1	-	**21**	**3**
C_2	C_2	2	4	-	-	-	1	-	-	-	-	**2**	**5**
C_3	C_3	2	2	2	-	3	-	2	-	6	-	**15**	**2**
C_4	C_4	2	2	2	-	3	-	2	-	6	-	**15**	**2**
C_5	C_5	1	3	-	2	-	1	-	-	-	-	**1**	**6**
C_6	C_6	-	1	-	-	3	-	-	1	-	-	**3**	**2**
C_7	C_7	-	1	6	1	1	-	5	-	1	-	**13**	**2**
C_8	C_8	-	-	-	-	2	3	2	-	2	-	**6**	**3**
C_9	C_9	-	-	-	-	3	3	-	-	-	-	**3**	**3**
C_{10}	C_{10}	-	-	-	3	1	2	-	4	-	-	**1**	**9**
C_{11}	C_{11}	-	-	-	-	1	3	-	-	-	4	**1**	**7**
C_{12}	C_{12}	-	-	1	-	-	4	1	-	-	-	**2**	**4**
C_{13}	C_{13}	-	-	1	-	-	1	3	-	-	-	**4**	**1**
C_{14}	C_{14}	-	-	1	1	-	1	-	1	1	-	**2**	**3**
C_{15}	C_{15}	-	-	1	-	-	1	1	-	3	1	**5**	**2**
C_{16}	C_{16}	-	-	-	2	-	-	1	1	1	-	**2**	**3**
C_{17}	C_{17}	-	-	-	3	-	-	1	-	-	-	**1**	**3**
C_{18}	C_{18}	-	-	-	1	-	-	1	4	-	-	**1**	**5**
C_{19}	C_{19}	-	-	-	3	-	-	1	-	-	-	**1**	**3**
C_{20}	C_{20}	-	-	-	1	-	-	1	1	-	-	**1**	**2**
C_{21}	C_{21}	-	-	-	1	-	-	-	2	2	-	**2**	**3**
C_{22}	C_{22}	-	-	-	1	-	-	-	1	2	-	**2**	**2**
C_{23}	C_{23}	-	-	-	-	-	-	-	1	1	1	**1**	**2**
-	C_{24}	-	-	-	-	-	-	-	1	-	-	-	**1**
-	C_{25}	-	-	-	-	-	-	-	1	-	-	-	**1**
-	C_{26}	-	-	-	-	-	-	-	1	-	2	-	**3**
-	C_{27}	-	-	-	-	-	-	-	1	-	-	-	**1**
-	C_{28}	-	-	-	-	-	-	-	-	-	2	-	**2**
-	C_{29}	-	-	-	-	-	-	-	-	-	3	-	**3**
-	C_{30}	-	-	-	-	-	-	-	-	-	2	-	**2**
-	C_{31}	-	-	-	-	-	-	-	-	-	1	-	**1**
-	C_{32}	-	-	-	-	-	-	-	-	-	2	-	**2**
Total		**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**100**	**100**

6 Conclusion

This work proposed a text categorization approach using fuzzy similarity to improve the issue of linguistic ambiguities found in text classification and using agglomerative hierarchical method to create categories based on the similarity analysis of textual terms.

Table 9 Texts clustered by Eurekha using the Star and Cliques algorithm in the TeMário Corpus. *Folha de São Paulo* is abbreviated as FSP and *Jornal do Brasil* is abbreviated as JB.

Source Text with Origin and Title													
Categories		**FSP**						**JB**				Total	
Created		**Opinion**		**World**		**Special**		**Intern.**		**Politics**		Categories	
S	C	S	C	S	C	S	C	S	C	S	C	S	C
C_1	**C_1**	10	-	9	-	10	2	3	-	1	-	**33**	**2**
C_2	**C_2**	2	-	3	-	2	3	5	-	1	-	**13**	**3**
C_3	**C_3**	-	-	2	-	4	2	2	-	10	-	**18**	**2**
C_4	**C_4**	-	-	3	-	1	4	2	-	-	-	**6**	**4**
C_5	**C_5**	1	-	1	-	2	2	-	-	1	-	**5**	**2**
C_6	**C_6**	-	-	1	-	-	3	7	-	2	-	**10**	**3**
C_7	**C_7**	1	-	-	-	-	3	1	2	1	-	**3**	**5**
C_8	**C_8**	-	-	1	-	-	-	-	5	1	-	**2**	**5**
C_9	**C_9**	4	-	-	-	-	-	-	2	-	-	**4**	**2**
C_{10}	**C_{10}**	2	-	-	-	1	-	-	3	3	-	**6**	**3**
-	**C_{11}**	-	1	-	1	-	-	-	1	-	-	-	**3**
-	**C_{12}**	-	-	-	1	-	-	-	5	-	-	-	**6**
-	**C_{13}**	-	-	-	-	-	-	-	2	-	-	-	**2**
-	**C_{14}**	-	-	-	2	-	-	-	-	-	-	-	**2**
-	**C_{15}**	-	-	-	2	-	-	-	-	-	-	-	**2**
-	**C_{16}**	-	-	-	2	-	-	-	-	-	-	-	**2**
-	**C_{17}**	-	-	-	2	-	-	-	-	-	-	-	**2**
-	**C_{18}**	-	-	-	2	-	-	-	-	-	-	-	**2**
-	**C_{19}**	-	2	-	2	-	-	-	-	-	-	-	**4**
-	**C_{20}**	-	-	-	2	-	-	-	1	-	-	-	**2**
-	**C_{21}**	-	1	-	2	-	-	-	-	-	-	-	**3**
-	**C_{22}**	-	3	-	1	-	-	-	-	-	-	-	**4**
-	**C_{23}**	-	-	-	1	-	-	-	-	-	2	-	**3**
-	**C_{24}**	-	2	-	-	-	-	-	-	-	-	-	**2**
-	**C_{25}**	-	2	-	-	-	-	-	-	-	-	-	**2**
-	**C_{26}**	-	4	-	-	-	-	-	-	-	-	-	**4**
-	**C_{27}**	-	2	-	-	-	-	-	-	-	-	-	**2**
-	**C_{28}**	-	2	-	-	-	-	-	-	-	-	-	**2**
-	**C_{29}**	-	1	-	-	-	-	-	-	-	3	-	**4**
-	**C_{30}**	-	-	-	-	-	-	-	-	-	5	-	**5**
-	**C_{31}**	-	-	-	-	-	-	-	-	-	4	-	**4**
-	**C_{32}**	-	-	-	-	-	-	-	-	-	2	-	**2**
-	**C_{33}**	-	-	-	-	-	1	-	-	-	4	-	**5**
Total		**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**20**	**100**	**100**

The technique of relative frequency adopted in the selection step allows the Categorizer to have the lists of the words that appear most often in the text. This technique was imperative in order to indicate which terms within the collection have a higher level of significance; that is, it established a threshold to decrease the dimensionality of the characteristics' vector space. The fuzzy similarity technique (set theoretic inclusion) used in the Categorizer, determined the inference function of the fuzzy logic, thereby allowing us to measure the similarity between the texts on the list. Star and Cliques algorithms were employed in the Agglomerative Hierarchical

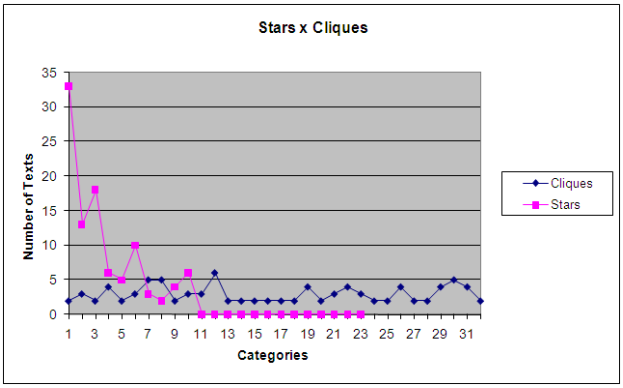

Fig. 11 Comparison between the stars and Cliques algorithm using Eurekha on TeMário corpus.

methodology to identify the groups of texts by specifying some type of relationship rule. Although they obtained very similar results, the Cliques algorithm presented a slight advantage of the Stars algorithm in that it created a larger number of clusters.

With regards to the comparison between the two categorizers that were studied, Eurekha and the Categorizer, there is statistical evidence suggesting that the Categorizer is more significant that Eurekha. In all of the simulations conducted, Eurekha attained a higher number of created categories in only 16.66% of the corpuses. Even so, the difference between the categories created by Eurekha and the Categorizer was of only one unit, in only one corpus and using only the Cliques algorithm.

In the Reuters corpus the Eurekha categorizer had its worst performance: when using the Star algorithm, Eurekha obtained only two categories, which made it impossible to carry on any sort of analysis looking to assess the relationship between the clustered texts.

On the other hand, the results with the TeMário corpus were very interesting, due to the fact that the corpus was developed with the purpose of summarizing texts that are very close and that are divided into previously established categories, which considerably facilitates the content analysis as well as the treatment of the file names.

The results of Eurekha and the Categorizer showed a very close proximity, up to the point where they were literally equal to one another. However, with the use of the F-test, it was seen that the variances were in fact different. As a subjective means of evaluation, we also verified a considerable advantage of the Categorizer in comparison to the Eurekha by comparing the results obtained with results from a human evaluator. The categorization of the Categorizer was much closer to what was considered ideal by the human evaluator.

Another interesting result was that of the RSS_Terra corpus, which in the F-test had its null hypothesis accepted. From that point, the t-Student test was conducted for equal variances after which the means from the experiments were evaluated. In the t-Student test, the hypothesis was also accepted. However, the critical-p was greater than its significance value α, which does not prove there was statistical evi-

dence that the means were equal. To conclude this work, the methodology proposed in this study points to encouraging results. The combination of the proposed techniques in each step of the Categorizer was very important in order for the results to be able to reach a positive indicative level.

One of the greatest challenges in this field is the reduction of vector space; that is, the calculation of the similarity matrix. As a proposal for the future, we expect to exclude these calculations, thereby reducing vector space as well as reducing the computational complexity of the text categorization algorithms.

Another common problem in the field is the definition of the stop criterion, which still stands in the way of a truly autonomous process. A common practice is to establish these criteria based on observations of the classifier's behavior. Note however that this problem is quite serious from the viewpoint of knowledge discovery, since this scenario is made up of groups of texts that are considerably dense and lengthy. A future contribution could be to use an automatic learning process to make decisions on a variety of circumstances regarding the best stop criterion to be used.

References

1. M. S. Aldenderfe, Roger K. Mark, and S. Aldenderfe. *Cluster Analysis*, page 88. SAGE University Publications (USA), Beverly Hills, 1978.
2. R. Arora and P. Bangarole. Text mining: classification & clustering of articles related to sports. In *Proceedings of the 43rd annual Southeast regional conference - Volume 1 ACM-SE 43*. ACM Press, 2005.
3. P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.
4. V. Cross. Fuzzy information retrieval. *Journal of Intelligent Information Systems*, 3:29–56, 1994.
5. I. Dagan, R. Feldman, and H. Hirsh. Keyword-based browsing and analysis of large document sets. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval — SDAIR*, pages 191–208, Las Vegas - Nevada, 1996.
6. B. S. Everitt and G. Dunn. *Applied Multivariate Data Analysis*. Edward Arnold, London, 2 edition, 2000. `http://www.iop.kcl.ac.uk/iop/Departments/BioComp/MvBook.stm`.
7. D. Fasulo. An analysis of recent work on clustering algorithms. Technical report, Univ. of Washington, Washington D.C., 1999. Technical Report.
8. U. Fayyad and R. Uthurusamy. *Data mining and knowledge discovery in databases: Introduction to the special issue Book: Editorial. Data Mining and Knowledge Discovery*. ACM, 1999.
9. R. Feldman and Haym Hirsh. Exploiting background information in knowledge discovery from text. *Journal of Intelligent Information Systems*, 1, 1997.
10. W. J. Frawley, S. G. Piatestsky, and C. Matheus. Knowledge discovery in data bases: an overview. AI Magazine. Fall 1992, 1992. `http://www.kdnuggets.com/gpspubs/aimag-kdd-overview-1992.pdf`.
11. L. Fávero. *Coesão e Coerência Textuais (in Portuguese)*. Ática, São Paulo, 2000.
12. J. Han and M. Kamber. *Data Mining: Concepts and Techiniques*. Morgan Kaufmann Publishers, New York, 1 edition, 2001.
13. M. A. Hearst. *Automated discovery of wordnet relations*. MIT Press, University Cambridge, 1998.
14. M. Hellmann. Fuzzy logic introduction. Université de Rennes, 2001. Journal.

15. A. K. Jain and R. C. Dubes. Algorithms for clustering data. Prentice Hall, 1988. http://www.cse.msu.edu/~jain/Clustering_Jain_Dubes.pdf.
16. W. Jianan and A. Rangaswamy. A fuzzy set model of consideration set formation calibrated on data from an online supermarket. EBusiness research Center Working Paper, n. 5, 1999.
17. G. Karypis and S. H. E. Han. Chameleon: Hierarchical clustering using dynamic modeling, 1999.
18. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration, 2002.
19. G. J. Klir and T. A. Folger. *Fuzzy sets, uncertainty, and information*. Prentice Hall, New Jersey, 1988.
20. G. Kowalski. Information retrieval systems: Theory and implementatio, 1997.
21. R.C. Kwok, J. Ma, and D. Zhou. Improving group decision making: A fuzzy gss approach. *IEEE Transactions on Systems, Man, and Cybernatics — Part C: Applications and Reviews*, 32:54–63, 2002.
22. T. M. Mitchell. Machine learning. McGraw-Hill Series in Computer Science, 1997.
23. S. Mitra and T. Acharya. *Data Mining: multimedia, soft computing, and bioinformatics*. Winley, 2003.
24. J. Moscarola and R. Bolden. From the data mine to the knowledge mill: applying the principles of lexical analysis to the data mining and knowledge discovery process. Technical report, Université de Savoie, 1998.
25. H. M. Oliveira. Seleção de entes complexos usando lógica difusa. Dissertation (Masters in Computer Science) - Instituto de Informática, 1996. In Portuguese.
26. T. A. S. Pardo. Dmsumm: Um gerador automático de sumários. Master's thesis, Universidade Federal de São Carlos, São Carlos, 2002. In Portuguese.
27. W. M. Pottenger and T. Yang. Dmsumm: Um gerador automático de sumários. (in Portuguese). Detecting emerging concepts in textual data mining. In: Michael Berry (ed.), Computational Information Retrieval, SIAM, Philadelphia, August 2001, 2001.
28. F.J. Rohf and R. R. Sokal. Statistical tables, 2nd ed.,usa, 1981.
29. C. Yen S. Tsaur, T. Chang. The evaluation of airline service quality bu fuzzy mcdm. Tourism Management, n. 23. Available at: <http://mslab.hau.ac.kr/mgyoon/master_02/ahp8.pdf>. Accessed on June 23, 2007, 2002. Lecture Notes in Computer Science, 1574.
30. G. Salton. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
31. C. M. Silva, M. C. Vidigal, P. S. Vidigal Filho, C. A. Scapim, E. Daros, and L. Silvério. Genetic diversity among sugarcane clones (saccharum spp.). *Scientiarum. Agronomy, v.27*, pages 315–319, 2005.
32. G. W. Snedecor. Calculation and interpretation of analysis of variance and covariance, 1934.
33. A. H. Tan. Text mining: the state of the art and the challenges. In *Workshop on knowledge discovery from advanced databases*, pages 65–70. Lecture Notes in Computer Science, 1999.
34. S. Velickov. Textminer theoretical background. http://www.delft-cluster.nl/textminer/theory/. Acessed on: Sep. 10th, 2007, 2004.
35. D. S. Vianna. Heurísticas híbridas para o problema da logenia. (in Portuguese). PhD Thesis. Pontifícia Universidade Católica - PUC, Rio de Janeiro, Brazil, 2004.
36. I.H. Witten, A. Moffat, and T. C. Bell. Managing gigabytes. Van Nostrand Reinhold. New York, 1994.
37. L. K. Wives. Um estudo sobre agrupamento de documentos textuais em processamento de informações não estruturadas usando técnicas de clustering. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, 1999. In Portuguese.
38. L. K. Wives. Utilizando conceitos como descritores de textos para o processo de identificação de conglomerados (clustering) de documentos. (in Portuguese). PhD. Thesis - Universidade Federal do Rio Grande do Sul.Programa de Pós-graduação em Computação, Porto Alegre, RS, Brazil, 2004.
39. L. K. Wives and N. A. Rodrigues. Eurekha. Revista Eletrônica da Escola de Administração da UFRGS (READ), Porto Alegre, v.6, n.5, 2000. In Portuguese.
40. L. A. Zadeh. Fuzzy sets. information and control. [S.l.], v.8, 1965.
41. L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Transactions on Systems, Man and Cybernetics v.SMC-3*, pages 28–44, 1973.