CHOOSING A DISTANCE METRIC FOR AUTOMATIC WORD CATEGORIZATION

Emin Erkan Korkmaz **Göktürk Üçoluk**

Department of Computer Engineering
Middle East Technical University
Ankara-Turkey

Emails: korkmaz@ceng.metu.edu.tr
ucoluk@ceng.metu.edu.tr

Abstract

This paper analyzes the functionality of different distance metrics that can be used in a bottom-up unsupervised algorithm for automatic word categorization. The proposed method uses a modified greedy-type algorithm. The formulations of fuzzy theory are also used to calculate the degree of membership for the elements in the linguistic clusters formed. The unigram and the bigram statistics of a corpus of about two million words are used. Empirical comparisons are made in order to support the discussions proposed for the type of distance metric that would be most suitable for measuring the similarity between linguistic elements.

1 Introduction

Statistical natural language processing is a challenging area in the field of computational natural language learning. Researchers of this field have an approach to language acquisition in which learning is visualized as developing a generative, stochastic model of language and putting this model into practice (Marcken, 1996).

Automatic word categorization is an important field of application in statistical natural language processing where the process is unsupervised and is carried out by working on n-gram statistics to find out the categories of words. Research in this area points out that it is possible to determine the structure of a natural language by examining the regularities of the statistics of language (Finch, 1993).

It is possible to construct a bottom-up unsupervised algorithm for the categorization process. In our paper named "A Method for Improving Automatic Word Categorization"(Korkmaz&Üçoluk, 1997) such a method, using a modified greedy-type algorithm supported by the notions of fuzzy logic, has been proposed. The distance metric used to measure the similarities of linguistic elements in this research is the *Manhattan Metric*. This metric is based on the absolute difference between the corresponding values of vector components. The components of the vectors correspond to bigram statistics of words for our case. However words from the same linguistic category in natural language may have totally different frequencies. So using a distance metric based on only the absolute differences may not be so suitable for the linguistic categorization process. In this paper various distance metrics are analyzed with the same algorithm in order to find out the most suitable one that could be used for linguistic elements. Comparisons are made for the results obtained using different metrics.

The organization of this paper is as follows. First the related work in the area of word categorization is presented in section 2. Then a general description of the categorization process and our proposed algorithm is given in 3 section, which is followed by presentation of different distance metrics that can be used with the algorithm. In section 5 the results of the experiments and the comparisons between the metrics are given. We discuss the relevance of the results and conclude in the last section.

2 Related Work

Usually unigram and the bigram statistics are used for automatic word categorization. There exists research where bigram statistics are used for the determination of the weight matrix of a neural network (Finch, 1992). Also bigrams are used with greedy algorithm to form the hierarchical clusters of words (Brown, 1992).

Genetic algorithms have also been successfully used for the categorization process(Lankhorst, 1994). Lankhorst uses genetic algorithms to determine the members of predetermined classes. The drawback of his work is that the number of classes is determined previous to run-time and the genetic algorithm only searches for the membership of those classes.

McMahon and Smith also use the bigram statistics of a corpus to find the hierarchical clusters (McMahon, 1996). However instead of using a greedy algorithm they use a top-down approach to form the clusters. Firstly the system divides the initial set containing all the words to be clustered into two parts and then the process continues on these new clusters iteratively.

Statistical NLP methods have been used also together with other methods of NLP. Wilms (Wilms, 1995) uses corpus based techniques together with knowledge-based techniques in order to induce a lexical sublanguage grammar. Machine Translation is an other area where knowledge bases and statistics are integrated. Knight et al., (Knight, 1994) aim to scale-up grammar-based, knowledge-based MT techniques by means of statistical methods.

3 Word Categorization

Zipf, (Zipf, 1935), who is a linguist, was one of the early researchers in statistical language models. His work states that 66% of large English corpus will fall within the first $2,000$ most frequent words. Therefore, the number of distinct structures needed to find an approximation to a large proportion of natural language would be small compared to the size of corpus that could be used. It can be claimed that by working on a small set consisting of frequent words, it is possible to build a framework for the whole natural language.

N-gram models of language are commonly used to build up such a framework. An N-gram model can be formed by collecting the probabilities of word streams $\langle w_i | i = 1..n \rangle$ where w_i is followed by w_{i+1}. These probabilities will be used to form the model where we can predict the behavior of the language up to n words. There exists current research that uses bigram statistics for word categorization in which probabilities of word pairs in the text are collected and processed.

These n-gram models can be used together with the concept of mutual information to form the clusters. *Mutual Information* is based on the concept of *entropy* which can be defined informally as the unpredictability of a stochastic experiment. For linguistic categorization, mutual information calculated would denote the amount of knowledge preserved in the bigram statistics. The detailed explanation of mutual information and adapting the formulations for automatic word categorization process could be found in (Lankhorst, 1994).

3.1 Clustering Approach

When the mutual information is used for clustering, the process is carried out somewhat at a macro-level. Usually search techniques and tools are used together with the mutual information in order to form some combinations of different sets, each of which is then subject to some *validity test*. The idea used for the validity testing process is as follows. Since the mutual information denotes the amount of probabilistic knowledge that a word provides on the proceeding word, if similar behaving words would be collected together into the same cluster, then the loss of mutual information would be minimal. So, the search is among possible alternatives for sets or clusters with the aim to obtain a minimal loss in mutual information.

Though this top-to-bottom method seems theoretically possible, in the presented work (Korkmaz&Üçoluk, 1997) a different approach, which is bottom-up, is used. In this incremental approach, set prototypes are built and then combined with other sets or single words to form larger ones. The method is based on the similarities or differences between single words rather than the mutual information of a whole corpus. In combining words into sets a *fuzzy set* approach is used.

Using this constructive approach, it is possible to visualize the word clustering problem as the problem of clustering points in an n-dimensional space if the lexicon space to be clustered consists of n words. The points which are the words of the corpus are positioned on this n-dimensional space according to their behavior relative to other words in the lexicon space. Each word is placed on the i^{th} dimension according to its bigram statistic with the word representing the dimension namely w_i. So the degree of *similarity* between two words can be defined as having close bigram statistics in the corpus. Words are distributed in the n-dimensional space according to those bigram statistics. The idea is quite simple: Let w_1 and w_2 be two words from the corpus. Let Z be the stochastic variable ranging over the words to be clustered. Then if $P_X(w_1, Z)$ is close to $P_X(w_2, Z)$ and if $P_X(Z, w_1)$ is close to $P_X(Z, w_2)$ for Z ranging over all the words to be clustered in the corpus, then we can state a *closeness* between the words w_1 and w_2. Here P_X is the probability of occurrences of word pairs. $P_X(w_1, Z)$ is the probability where w_1 appears as the first element in a word pair and $P_X(Z, w_1)$ is the reverse probability where w_1 is the second element of the word pair. This is the same for w_2 respectively.

In order to start the clustering process, a distance function has to be defined between the elements in the space. Assume that the bigram statistics for word couples are placed in a matrix N, where N_{ij} denotes the number of times word-couple (w_i, w_j) is observed in the corpus. So formulating the similarity between two linguistic elements would be finding out the distance between two vectors that can be obtained from this matrix. Different distance metrics are proposed for the distance between vectors. The usage of a distance metric forms the main discussion point of this paper. In next section first the algorithm used for categorization will be presented and in section 4 these metrics and their usage for linguistic categorization will be discussed.

3.2 The Algorithm for Categorization

Having a distance function, it is possible to start the clustering process. The first idea that can be used is to form a greedy algorithm to start forming the hierarchy of word clusters. If the lexicon space to be clustered consists of $\{w_1, w_2, ..., w_n\}$, then the first element from the lexicon space w_1 is taken and a cluster with this word and its nearest neighbor or neighbors is formed. Then the lexicon space is $\{(w_1, w_{s_1}, ..., w_{s_k}), w_i, ..., w_n\}$ where $(w_1, w_{s_1}, ..., w_{s_k})$ is the first cluster formed. The process is repeated with the first element in the list which does not belong to any set yet (w_i for our case) and the process iterates until no such word is left. The sets formed will be the clusters at the bottom of the cluster hierarchy. Then to determine the behavior of a set, the frequencies of its elements are added and the previous process this time is carried on the sets rather than on single words until the cluster hierarchy is formed, so the algorithm stops when a single set is formed that contains all the words in the lexicon space.

In the early stages of this research such a greedy method was used to form the clusters. However, though some clusters at the low levels of the tree seemed to be correctly formed, as the number of elements in a cluster increased towards the higher levels, the clustering results became unsatisfactory.

Two main factors were observed as the reasons for the unsatisfactory results.

These were:

- Shortcomings of the greedy type algorithm.
- inadequacy of the method used to obtain the set behavior from the properties of its elements.

The greedy method results in a non optimal clustering in the initial level. To make this point clearer consider the following example: Let us assume that four words w_1,w_2, w_3 and w_4 are forming the lexicon space. Furthermore, let the distances between these words be defined as d_{w_i,w_j}. Then consider the distribution in Figure 1. If the greedy method first tries to cluster w_1, then it will be clustered with w_2, since the smallest d_{w_1,w_i} value is d_{w_1,w_2}. So the second word will be captured in the set and the algorithm will continue the clustering process with w_3. At this point, though w_3 is closest to w_2, it is captured in a set and since w_3 is closer to w_4 than the center of this set is, a new cluster will be formed with members w_3 and w_4. However, as it can be obviously seen visually from Figure 1 the first optimal cluster to be formed between these four words is the set $\{w_2, w_3\}$.

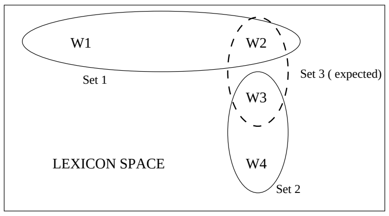

Figure 1: Example for the clustering problem of greedy algorithm in a lexicon space with four different words. Note that d_{w_2,w_3} is the smallest distance in the distribution. However since w_1 is taken into consideration, it forms set1 with its nearest neighbor w_2 and w_3 combines with w_4 and form set2, although w_2 is nearer. And the expected third set is not formed.

The second problem causing unsatisfactory clustering occurs after the initial sets are formed. According to the algorithm, the clusters behave exactly like other single words and participate in the clustering just as single words do. However to continue the process, the bigram statistics of the clusters should be determined. This means that the distance between the cluster and all the other elements in the search space have to be calculated. One easy way to determine this behavior is to find the average of the statistics of all the elements in a cluster. This method has its drawbacks. If the corpus used for the process is not large, the proximity problem becomes severe. On the other hand the linguistic role of a word may vary in contexts in different sentences. Many words are used as noun, adjective or falling into some other linguistic category depending on the context. It can be claimed that each word initially shall be placed in a cluster according to its dominant role. However to determine the behavior of a set the dominant roles of its elements should also be used. Somehow the common properties (bigrams) of the elements should be always used and the deviations of each element should be eliminated in the process.

3.2.1 Improving the Greedy Method

The clustering process is improved to overcome the above mentioned drawbacks. To overcome the first problem the idea used is to allow words to be members of more than one cluster. So after the first pass over the lexicon space, intersecting clusters are formed. For the lexicon space presented in Figure 1 with four words, the expected third set will be also formed. As the second step these intersecting sets are combined into a single set. Then the closest two words in each combined set (according to the distance function) are found and these two closest words are taken into consideration as the centroid for that set. After finding the centroids of all sets, the distances between a member and all the centroids are calculated for all the words in the lexicon space. Following this, each word is moved to the set where the distance between this member and the set center is minimal. This procedure is necessary since the initial sets are formed by combining the intersecting sets. When these intersecting sets are combined the set center of the resulting set might be far away from some elements and there may be other closer set centers formed by other combinations, so a reorganization of membership is appropriate.

3.2.2 Fuzzy Membership

As presented in the previous section the clustering process builds up a cluster hierarchy. In the first step, words are combined to form the initial clusters, then those clusters become members of the process themselves. To combine clusters into new ones their statistical behavior should be determined. The statistical behavior of a cluster is related to the bigrams of its members. In order to find out the dominant statistical role of each cluster the notion of fuzzy membership is used.

The problem that each word can belong to more than one linguistic category brings up the idea that the sets of word clusters cannot have crisp border lines and even if a word seems to be in a set due to its dominant linguistic role in the corpus, it can have a degree of membership to the other clusters in the search space. Therefore the concept of fuzzy membership can be used for determining the bigram statistics of a cluster.

Researchers working on fuzzy clustering present a framework for defining fuzzy membership of elements. Gath and Geva (Gath, 1989) describe such an unsupervised optimal fuzzy clustering. They present the K-means algorithm based on minimization of an objective function. For the purpose of this research only the membership function of the algorithm presented is used. The membership function u_{ij} that is the degree of membership of the i^{th} element to the j^{th} cluster is defined as:

$$u_{ij} = \frac{\left| \frac{1}{d^2(X_i, V_j)} \right|^{\frac{1}{(q-1)}}}{\sum_{k=1}^{K} \left| \frac{1}{d^2(X_i, V_j)} \right|^{\frac{1}{(q-1)}}} \qquad (1)$$

Here X_i denotes an element in the search space, V_j is the centroid of the j^{th} cluster. K denotes the number of clusters. And $d^2(X_i, V_j)$ is the distance of $X_i th$ element to the centroid V_j of the j^{th} cluster. The parameter q is the weighting exponent for u_{ij} and controls the fuzziness of the resulting cluster.

After the degrees of membership of all the elements of all classes in the search space are calculated, the bigram statistics of the classes are derived. To find those statistics the following method is used: For each subject cluster, the bigram statistics of each element is multiplied with its membership value. This forms the amount of statistical knowledge passed from the element to that set. So the elements chosen as set centroids will be the ones that affect a set's statistical behavior the most. Hence an element away from a centroid will have a lesser statistical contribution.

4 Distance Metrics

Various distance metrics have been proposed by mathematicians that can be used to formulate the similarity between vectors. Four of them are examined and used for this study. The first one is the *Manhattan Metric* which just calculates the absolute difference between the values of two vector elements. It is defined by:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{1 \leq i \leq n} \mid x_i - y_i \mid \qquad (2)$$

Here $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and $\mathbf{y} = \{y_1, y_2, ..., y_n\}$ are two vectors defined over $\mathcal{R}^n$.

Having such a metric it is possible to define the distance function between two linguistic elements. The distance function D between two words w_1 and w_2 could be defined as follows:

$$D(w_1, w_2) = D_1(w_1, w_2) + D_2(w_1, w_2) \qquad (3)$$

Here the distance function consists of two different parts D_1 and D_2. This is because we want the distance function to be based on both proceeding and preceding words. So the first part denotes the distance on proceeding words and the second one denotes the distance obviously on the preceding words. If we use the *Manhattan metric*, the distance function would be :

$$D(w_1, w_2) = \sum_{1 \leq i \leq n} \mid N_{w_1 i} - N_{w_2 i} \mid + \mid N_{i w_1} - N_{i w_2} \mid \tag{4}$$

Here n is the total number of words to be clustered, $N_{w_1 i}$ is the number of times word couple (w_1, w_i) is observed in the corpus and $N_{i w_1}$ is the number of times word couple (w_i, w_1) is observed. Obviously it is the same for word w_2. This distance metric just calculates the total difference on two vector-couples obtained from the frequency matrix N, where the first couple denotes the vectors obtained by the frequencies of the word-couples formed by w_1, w_2 and their proceeding words. The second couple denotes the vectors formed by the frequencies with the preceding words correspondingly.

The above formulation explains the structure of the distance metric used for the study. For the researched presented in our previous paper (Korkmaz&Üçoluk, 1997) *Manhattan Metric* was the only metric used for the distance function. However others are proposed for the similarity between vectors. Another metric is the *Euclidean Metric*:

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{1 \leq i \leq n} (x_i - y_i)^2} \tag{5}$$

Here $\mathbf{x}$ and $\mathbf{y}$ are again two vectors defined over $\mathcal{R}^n$. Also the formulation of the angle between two vectors is also used for this study as a distance metric. If θ is the angle between the two vectors $\mathbf{x}$ and $\mathbf{y}$, then $\cos\theta$ is calculated by:

$$\cos\theta = \frac{\mathbf{x}'\mathbf{y}}{\mid \mathbf{x} \mid\mid \mathbf{y} \mid} = \frac{\sum_{1 \leq i \leq n} x_i y_i}{[\sum_{1 \leq i \leq n} {x_i}^2]^{\frac{1}{2}} [\sum_{1 \leq i \leq n} {y_i}^2]^{\frac{1}{2}}} \tag{6}$$

Here, $\mathbf{x}'\mathbf{y}$ denote the scalar product of the two vectors $\mathbf{x}$ and $\mathbf{y}$ and $\mid \mathbf{x} \mid$ denote the magnitude of the vector $\mathbf{x}$. Since the components of the vectors in our case are corresponding to the frequencies of words, they will be non-negative. So the angle between the two vectors will be between $0°$ and $90°$. Since $\cos 0°$ is unity and $\cos 90°$ is zero, a distance metric between the two vectors can be defined as:

$$D(\mathbf{x}, \mathbf{y}) = 1 - \cos\theta \tag{7}$$

This distance metric will give us a number from the closed interval $[0, 1]$, 0 denoting that the two vectors are overlapping and 1 denoting that there is an angle of $90°$ which is the highest difference between the vectors.

The last distance metric used for the similarity function is the *Spearman Rank Correlation Coefficient.* This metric is based on the difference between the *ranks* of two vectors rather than the difference between their elements. The metric is defined as:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{1 \leq i \leq n} ({R_i}^x - {R_i}^y)^2 \tag{8}$$

Here $\mathbf{x}$ and $\mathbf{y}$ are again two vectors as defined above. ${R_i}^x$ nd ${R_i}^y$ are the ranks of the corresponding vectors. The rank is calculated for our case by normalizing the vectors in the interval $[0, 1]$. The component with the highest value among the components of the vector takes the value 1 and if there are n elements in the vector, the one with the second highest value will correspond to the number $1-(1/n)$ and so on. The smallest value will correspond to zero.

For the process of formulating the distance between linguistic elements, the main problem appears due to the difference between the frequencies of words from the same linguistic category. For instance the word *go* has a very high frequency in natural language corpora compared to many other verbs, but still we have to cluster *go* with low frequency verbs. However if we use a distance metric based on only the absolute differences of vectors like the *Euclidean Metric* or *Manhattan Metric*, the distance calculated between high frequency and low frequency words would be high, which is undesired. Therefore when comparing a high frequency word with a low frequency one, we should be able to determine if the difference is caused by some regular magnitude difference. A similarity can exist between the corresponding values when this magnitude difference is discarded. Without having a distance function that compensates for this, it is not possible to overcome the errors introduced by having different frequencies for words from the same linguistic category. This acts as a considerable factor disturbing the quality of formed clusters.

Having this in mind the *Spearman Rank Correlation Coefficient Metric* and the *Angle Metric* are used as distance function. These two seem to discard the magnitude difference between the components of the vectors. Such a comparison seems to be more suitable for evaluating the similarity of linguistic elements.

In the *Spearman Rank Correlation Coefficient* the vectors are normalized into the closed interval $[0, 1]$. So the vectors are similar if the change from one component to the next is similar, regardless of the difference in the absolute values. We have a similar comparison for the *Angle Metric*. When this metric is used, the length of the two vectors compared may be totally different which reflects a fact of having totally different frequencies in natural language corpora. Regardless of the magnitude of the vectors, if the angle between the two vectors is small, they will be considered similar.

However the improvement obtained using these two metrics has not been so significant. It is believed that the two approaches, that is, comparing the bigram statistics absolutely and making somewhat a relative comparison between them are both effective in fetching out different aspects that lead to similarity between linguistic elements. Therefore, using just one of the approaches is not sufficient to increase the quality of formed clusters. Trying to combine these two approaches has lead to the fifth distance metric which is the combination of the *Angle Metric* and the *Euclidean Metric*.

This fifth metric which we call the *Combined Metric* is defined by:

$$D(\mathbf{x},\mathbf{y}) = (1-\cos\theta)\sqrt{\sum_{1\leq i\leq n}(x_i-y_i)^2} \qquad (9)$$

Here the $\cos\theta$ is the cosine of the angle between the two vectors. As explained above $(1-\cos\theta)$ will be a value from the closed interval $[0,1]$. The rest of the formulation gives us the Euclidean distance between the vectors. Multiplying this Euclidean distance by $(1-\cos\theta)$ has a following affect. If the angle between the vectors is small, meaning that the vectors are relatively similar, the result of the first part of the formulation will be close to zero. This will be a factor decreasing the value coming from the Euclidean distance. So even if the words have different frequencies in the corpora, their relative similarity will be a factor decreasing the absolute distance between them. With this formulation it is believed that both of the factors affecting the similarity between the linguistic categories are taken into account.

5 Results

The different distance metrics are again tested on the same corpus with the previous research. The corpus is formed with online novels collected from the WWW page of the "Book Stacks Unlimited, Inc." The corpus consists of twelve free on-line novels adding up to about $1,700,000$ words. The corpus is passed through a filtering process where the special words, useless characters and words are filtered out. After this initial process, frequencies of words are collected. The thousand most frequent words are chosen and sent to the clustering process. These most frequent thousand words form the 70.4% of the whole corpus. Percentage goes up to 77% if the next thousand most frequent words are added to the lexicon space. The first ten most frequent words in the corpora and their frequencies are presented in Table 1.

the	5.002056%
and	3.281249%
to	2.836796%
of	2.561952%
a	2.107116%
in	1.591189%
he	1.533916%
was	1.419838%
that	1.306431%
his	1.124362%
it	1.061797%

Table 1: Frequencies of the most frequent ten words expressed as a percentage of the total length of the corpus

The clustering process builds up a tree hierarchy having words at the leaves and clusters at the nodes. The root node denotes the largest class containing all the lexicon space. At each level a different cluster organization could be observed for the lexicon space. The leaves are the clusters of single words formed by the first pass of the algorithm. At the upper levels each node represents clusters formed as various combinations of clusters at the lower levels.

In the earlier research explained in the previous sections, the algorithm was only tested with the *Manhattan Metric*. According to the algorithm first initial clusters are formed and these initial clusters are combined into larger ones to form the cluster hierarchy. A significant convergence has been obtained with this metric with the initial clusters. However two main problems were encountered in the cluster hierarchy formed with this metric. The first one was a single large faulty initial cluster containing different linguistic categories. Although it is the nouns which are mainly located in this cluster, adjectives verbs and even prepositions are also added to it. This was a factor decreasing the success rate obtained in the initial clusters. It is believed that the cause for this faulty cluster is that the metric used is not so powerful. Remember that, according to the algorithm intersecting clusters are formed and they are combined into one cluster. If the distance metric used is not powerful enough, although we may get a very high success rate for some of clusters, certain clusters with different linguistic roles will seem to be overlapping and they might be combined into one cluster in the first pass of the algorithm. When the other metrics are used this large faulty cluster

Test Criteria	Manhattan Metric	Angle Metric	Euclidean Metric	Spearman Rank Correlation Coefficient	Combined Metric
# of initial clusters	60	169	132	185	171
# of elm. in the initial clusters	16.6	5.9	7.56	5.4	5.8
Depth of the tree	8	3	9	11	11
Location of leaves	$5^{th} and$ $6^{th} levels$	$3^{th} level$	$7^{th} and$ $8^{th} levels$	$9^{th} and$ $10^{th} levels$	$9^{th} and$ $10^{th} levels$
# of nodes on the second level	18	39	35	41	37

Table 2: Comparison of cluster hierarchies obtained with different metrics.

disappeared. We were able to get an initial success rate of about 90% with the *Manhattan Metric* when we discarded this large faulty cluster. However with the other metrics this success rate has been obtained for all the lexicon space.

The second problem encountered for the categorization process appears while combining the initial clusters into larger ones. Although it is possible to obtain some local successful combinations with the first metric, the overall performance in combining these initial clusters is not so satisfactory. So different metrics presented in section 4 have been tested on the algorithm. Unfortunately, although the proposed metrics were able to overcome the first problem of having a large faulty cluster, the progress obtained in combining initial clusters into larger ones was not so significant. This has been the factor triggering the idea that a metric taking into consideration both of the approaches for linguistic similarity would be more suitable for our case. So the fifth metric, the *Combined Metric*, has been constructed. The main progress obtained with this fifth metric is on the second problem described.

In table 2 the hierarchies obtained using different metrics are presented. When the properties presented in this table are examined, the hierarchy formed by the *Manhattan Metric* has the minimum number of initial clusters. This is due to the large faulty cluster formed with this metric. The properties of the hierarchies presented in table 2 seem to be similar to each other. Only the depth of the tree formed with the *Angle Metric* differs from the other ones. This is because more initial clusters are combined on the second level in the hierarchy obtained with this metric. This brings in an increase in the number of ill-structured clusters on the second level over-combining distinct linguistic categories.

5.1 Empirical Comparison

The main progress for the clustering hierarchy is obtained by the *Combined Metric*. It seems suitable to examine this metric in detail and compare the results with the initial organization obtained by the *Manhattan Metric*.

Some linguistic categories inferred by the algorithm using the *Combined Metric* are listed below:

- professor **opposite** church hall **least present once last** baby prisoner doctor wind gate village sun country
- earth forest garden truth river
- picture case glass
- captain servant book horse meeting situation *circumstances* summer afternoon evening night morning day future **early**
- large new small great very strange certain good fine few little
- slight **man's** sudden thousand hundred different
- rich fair **secret** blue soft cold bright quick frightened surprised plain clear true greater worse better tall dead living wrong
- notice cry hold touch influence act account form effect care
- meant **ought** wanted used **enough back** began tried turned came
- enter pass follow carry call give bring tell do let forgive
- impossible possible necessary
- calm pale warm simple sweet quiet busy hot angry ill
- aunt uncle sister **husband's**
- duty attention desire **turning coming close**
- listening **ready** trying going
- died fallen drawn learned written gone
- known taken brought given
- *shoulders* neck pocket hat chair shoulder arm mouth
- person girl lady woman gentleman man fellow else thing
- *affairs* age speech action marriage *questions ideas looks* silence society love experience
- between towards upon against after before like about round off away up
- under into through on at over
- during near toward beside within around behind **gave told took**
- shall should may will must would might **i**
- won't cannot can can't are didn't don't

	Combined Metric	Manhattan Metric
Nouns		
Largest # of words collected	94	111
Success Rate	91.5%	94.6%
# of initial clusters connected	15	6
Verbs (present perfect)		
Largest # of words collected	67	45
Success Rate	100%	73.3%
# of initial clusters connected	12	3
Verbs (past perfect)		
Largest # of words collected	16	2
Success Rate	100%	100%
# of initial clusters connected	5	1
Adjectives		
Largest # of words collected	68	17
Success Rate	92.6%	100%
# of initial clusters connected	7	2
Adverbs		
Largest # of words collected	9	4
Success Rate	100%	100%
# of initial clusters connected	1	1
Auxiliaries		
Largest # of words collected	7	9
Success Rate	100%	100%
# of initial clusters connected	1	1
Determiners		
Largest # of words collected	16	10
Success Rate	100%	100%
# of initial clusters connected	1	1

Table 3: Comparison made between Combined Metric and Manhattan Metric based on the largest number of elements combined in a cluster.

- anybody everyone nobody everybody everything
- **exactly** finding hearing watching **all** leaving seeing giving keeping knowing
- those these our an a this his their the your my her any no some not such its

The ill-placed members in the clusters above are shown using bold font. The above initial clusters represent the linguistic categories with a success rate of 90.2%. Also the plural nouns in singular noun clusters are shown in italics. If we consider those placements as faulty ones also, the calculated success rate would fall to 88.1%. This success rate seems to be similar to the results obtained with other distance metrics. However as explained above the main progress obtained with this *Combined Metric* is on the process of combining these initial clusters into larger ones in the upper levels of the cluster hierarchy.

Two examples from the cluster hierarchy obtained with this metric are given in tables 4 and 5. In table 4 94 nouns coming from different initial clusters are combined in the same part of the cluster hierarchy. Only one cluster seems to be misplaced in this region. This is an adjective cluster. In table 5 67 different verbs are collected. They are all present tense verbs and no misplaced word exists in this part of the hierarchy. This is another well-formed part of the cluster organization. It is believed that this is an important improvement compared to earlier results, since there is an increase in the number of successfully connected initial clusters.

Table 3 exhibits the improvement obtained using the *Combined Metric*. Maximum number of words correctly classified for some linguistic categories are shown in this table. Obviously there are other clusters having elements from the same linguistic categories in different parts of the hierarchy. This table makes a comparison of the maximum numbers of words successfully collected in order to analyze the improvement obtained. Gathering *nouns* and *auxiliaries* seems to be carried out better with the *Manhattan Metric*. However if we consider the number of initial clusters forming these largest ones, a significant progress seems to exist for the *Combined Metric*. There is a big difference for these numbers between the two. For instance 12 *present perfect verb* classes are combined successfully when the *Combined Metric* is used, but only 8 of them were combined with the *Manhattan Metric*. For *adjectives* this is 7 to 2, for *past perfect verbs* 5 to 1 and although number of *nouns* collected by the *Manhattan Metric* is larger, number of initial clusters successfully combined by the *Combined Metric* is still larger.

It can be claimed that there is a significant progress in the process of successfully combining the initial clusters when the new metric is used. This was the main problem encountered with the *Manhattan Metric* and the other ones. This is denoted as the progress obtained by using the *Combined Metric* trying to represent both of the two approaches that can be taken into account for the similarity of linguistic elements.

6 Discussion And Conclusion

This research has focussed on the usage of distance function for an unsupervised, bottom-up algorithm for automatic word categorization. The results obtained seem to show that natural language preserves the necessary information implicitly for the acquisition of the linguistic categories it has. A convergence of linguistic categories could be obtained by using the algorithm we have presented. This result is a motivating one for further studies on acquisition of

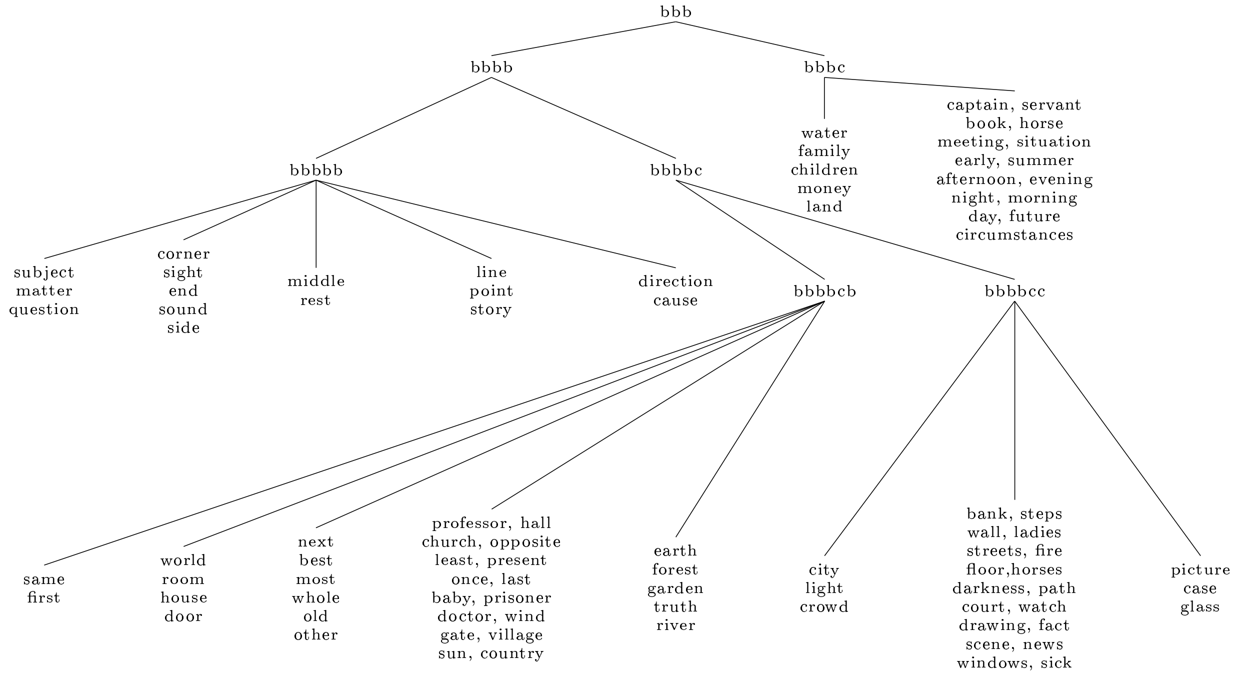

Table 4: Part of the cluster hierarchy holding nouns

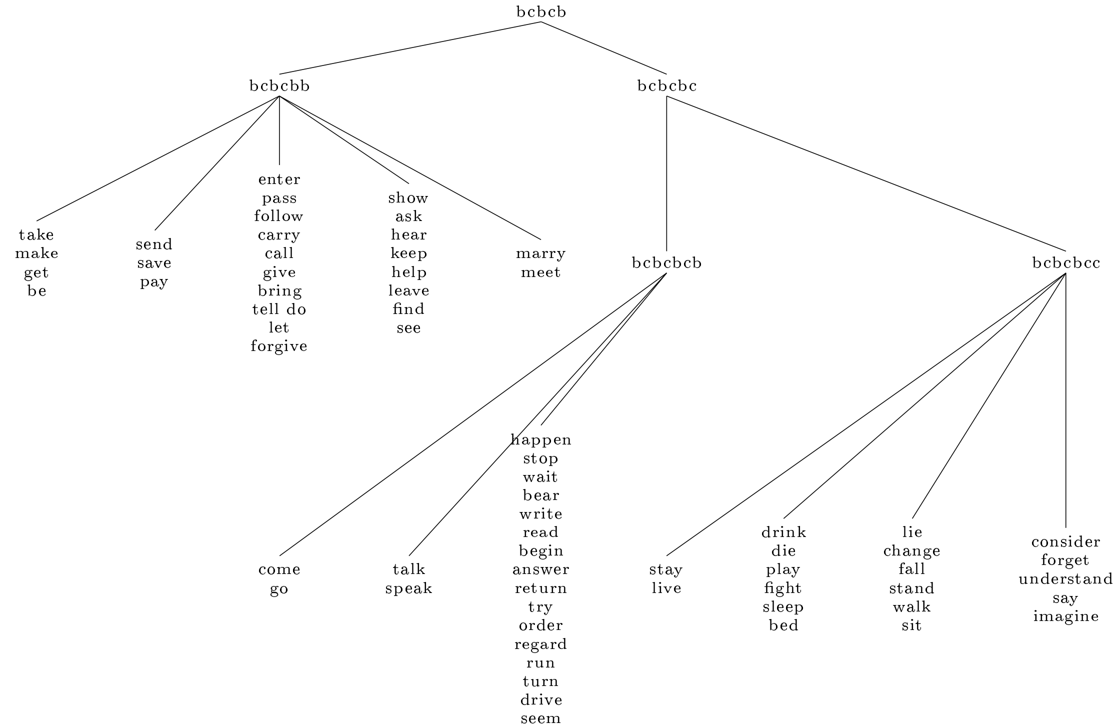

Table 5: Part of the cluster hierarchy holding present tense verbs

structures preserved in natural language at various abstraction levels.

Different distance metrics are used for the algorithm. The results obtained by the *Combined Metric* show that special distance metrics trying to combine different properties of linguistic elements could be developed for linguistic categorization.

Considering the results obtained by the experiments carried out, the following remarks could be made on the linguistic clusters formed in the study. In the initial clusters formed the success rate obtained is satisfactory. Though it was not possible to to combine these initial clusters into exact linguistic categories, the cluster hierarchy obtained with *Combined metric* is encouraging. The faulty placements are mainly due to the the very complex structure of natural language. The fact that many words can be used with different linguistic roles in natural language sentences produces deviations in the information given by the bigrams. Using fuzzy logic and a suitable distance metric is a way to decrease these deviations, however it was not possible to remove them totally.

References

Brown P.F., V.J. Della Pietra, P.V. deSouza, J.C. Lai, and R.L. Mercer. Class-based n-gram models of natural language. Computational Linguistics, 18(4):467-477, 1992

de Marcken, Carl G. Unsupervised Language Acquisition. PhD Thesis, Department of Electrical Engineering and Computer Science, Massachusetta Institute of Technology, 1996.

Finch, S. Finding Structure in language. PhD Thesis. Centre for Cognitive Science, University of Edinburgh, 1993.

Finch, S. and N. Chater, Automatic methods for finding linguistic categories. In Igor Alexander and John Taylor, editors, Artificial Neural Networks, volume 2. Elsevier Science Publishers, 1992.

Gath, I and A.B. Geva Unsupervised Optimal Fuzzy Clustering. IEEE Transactions on pattern analysis and machine intelligence, Vol. 11, No. 7, July 1989.

Knight, Kevin, Ishwar Chander, Haines Matthew, Hatzivassiloglou Vasieios, Hovy Eduard, Iida Masayo, Luk Steve, Okumura Akitoshi, Whitney Richard, Yamada Kenji. Intagrating Knowledge Bases and Statistics in MT. Proceedings of the 1st AMTA Conference. Columbia, MD. 1994.

Korkmaz, E. E. and G. Üçoluk A Method For Improving Automatic Word Categorization. Proceedings of the Workshop on Computational Natural Language Learning. (Conll97). Madrid, Spain. pp. 43-49, 1997.

Lankhorst, M.M. A Genetic Algorithm for Automatic Word Categorization. In: E. Backer (ed.), Proceedings of Computing Science in the Netherlands CSN'94, SION, 1994, pp. 171-182.

McMahon, John G. and Francis J. Smith. Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. Computational Linguistics, 22(2):217-247,1996.

Wilms, G. J. Automated Induction of a Lexical Sublanguage Grammar Using a Hybrid System of Corpus and Knowledge-Based Techniques. Mississippi State University. PhD Thesis, 1995.

Zipf, G.K. The psycho-biology of Language. Boston: Houghton Mifflin. 1935