A METHOD FOR IMPROVING AUTOMATIC WORD CATEGORIZATION

Emin Erkan Korkmaz **Göktürk Üçoluk**

Department of Computer Engineering
Middle East Technical University

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

Abstract

This paper presents a new approach to automatic word categorization which improves both the efficiency of the algorithm and the quality of the formed clusters. The unigram and the bigram statistics of a corpus of about two million words are used with an efficient distance function to measure the similarities of words, and a greedy algorithm to put the words into clusters. The notions of fuzzy clustering like cluster prototypes, degree of membership are used to form up the clusters. The algorithm is of unsupervised type and the number of clusters are determined at run-time.

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 visualised as developing a generative, stochastic model of language and putting this model into practice (de Marcken, 1996). It has been shown practically that the usage of such an approach can yield better performances for acquiring and representing the structure of language.

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 the language (Finch, 1993).

The organization of this paper is as follows. After the related work in the area of word categorization is presented in section 2, a general background of the categorization process is described in 3 section, which is followed by presentation of newly proposed method. In section 4 the results of the experiments are given. We discuss the relevance of the results and conclude in the last section.

2 Related Work

There exists previous work in which the unigram and the bigram statistics are used for automatic word clustering. Here it is concluded that the frequency of single words and the frequencies of occurance of word pairs of a large corpus can give the necessary information to build up the word clusters. Finch (Finch and Chater, 1992), makes use of these bigram statistics for the determination of the weight matrix of a neural network. Brown, (Brown et al., 1992) uses the same bigrams and by means of a greedy algorithm forms the hierarchical clusters of words.

Genetic algorithms have also been successfuly used for the categorization process. Lanchorst (Lankhorst, 1994) 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 (McMahon and Smith, 1996) also use the mutual information of a corpus to find the hierarchical clusters. However instead of using a greedy algorithm they use a top-down approach to form the clusters. Firstly by using the mutual information 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

Emin Erkan Korkmaz and Göktürk Üçoluk (1997) **A Method for Improving Automatic Word Categorization.** In T.M. Ellison (ed.) *CoNLL97: Computational Natural Language Learning,* ACL pp 43-49.

are integrated. Knight, (Knight et al., 1994) aims to scale-up grammar-based, knowledge-based MT techniques by means of statistical methods.

3 Word Categorization

The words in a natural language can be visualised as consisting of two different sets. The closed class words and the open class ones. New open class words can be added to the language as the language progresses, however the closed class is a fixed one and no new words are added to the set. For instance the prepositions are in the closed class. However nouns are in the open class, since new nouns can be added to the language as the social and economical life progresses. However the most frequently used words in a natural language usually form a closed class.

Zipf, (Zipf, 1935), who is a linguist, was one of the early researchers on statistical language models. His work states that only 2% of the words of a large English corpus form 66% of the total corpus. Therefore, 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 framework. An n-gram model can be formed by collecting the probabilities of word streams $\langle w_i | i = 1..n \rangle$. The probabilities will be used to form the model where we can predict the behaviour of the language up to n words. There exists current research that use bigram statistics for word categorization in which probabilities of word pairs in the text are collected and processed.

3.1 Mutual Information

As stated in the related work part 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 uncertainty of a stochastic experiment. Let X be a stochastic variable defined over the set $X = \{x_1, x_2, ..., x_n\}$ where the probabilities $P_X(x_i)$ are defined for $1 \leq i \leq n$ as $P_X(x_i) = P(X = x_i)$ then the entropy of X, $H(X)$ is defined as:

$$H\{X\} = -\sum_{1 \leq i \leq n} P_X(x_i) \log P_X(x_i) \qquad (1)$$

And if Y is another stochastic variable than the mutual information between these two stochastic variables is defined as:

$$I\{X : Y\} = H\{X\} + H\{Y\} - H\{X, Y\} \qquad (2)$$

Here $H\{X, Y\}$ is the joint entropy of the stochastic variables X and Y. The *joint entropy* is defined as:

$$H\{X, Y\} = -\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq m} P_{xy}(x_i, y_j) \log P_{xy}(x_i, y_j) \qquad (3)$$

And in this formulation $P_{xy}(x_i, y_j)$ is the *joint probability* defined as $P_{xy}(x_i, y_j) = P(X = x_i, Y = y_j)$

Given a lexicon space $W = \{w_1, w_2, ..., w_n\}$ consisting of n words to be clustered, we can use the formulation of mutual information for the bigram statistics of a natural language corpus. In this formulation X and Y are defined over the sets of the words appearing in the first and second positions respectively. So the mutual information that is the amount of knowledge that a word in a corpus can give about the proceeding word can be reformulated using the bigram statistics as follows:

$$I\{X : Y\} = \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} \frac{N_{ij}}{N_{**}} \log \frac{N_{ij}.N_{**}}{N_{i*}.N_{*j}} \qquad (4)$$

In this formulation N_{**} is the total number of word pairs in the corpus and N_{ij} is the number of occurances of word pair $\langle word_i, word_j \rangle$, N_{i*} is the number of occurences of $word_i$ and N_{*j} is the number of occurences of $word_j$ respectively. This formulation denotes the amount of linguistic knowledge preserved in bigram of words in a natural language.

3.2 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 in a corpus, if similar behaving words are collected together to the same clusters than 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.

Although this top-to-bottom method seems theoretically possible, in the presented work a different approach, which is bottom-up is used. In this incremental approach, set prototypes are first 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. The authors believe that this serves to determine the behavior of the whole set more properly.

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 that are the words in a corpus are positioned on this n-dimensional space according to their behaviour related to other words in the lexicon space. Each $word_i$ is placed on the i^{th} dimension according to its bigram statistic with the word representing the dimension. So the degree of *similarity* between two words can be defined as having close bigram statistics in the corpus. Words are distributed in the plane 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, than we can state a *closeness* between the words w_1 and w_2. Here P_X is the probability of occurences of word pairs as stated in section 3.1. $P_X(w_1, Z)$ 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 our plane. Using the idea presented above we define a simple distance function between words using the bigram statistics. The distance function D between two words w_1 and w_2 is defined as follows:

$$D(w_1, w_2) = D_1(w_1, w_2) + D_2(w_1, w_2) \qquad (5)$$

where

$$D_1(w_1, w_2) = \sum_{1 \le i \le n} \mid P_X(w_1, w_i) - P_X(w_2, w_i) \mid \qquad (6)$$

and

$$D_2(w_1, w_2) = \sum_{1 \le i \le n} \mid P_X(w_i, w_1) - P_X(w_i, w_2) \mid \qquad (7)$$

Here n is the total number of words to be clustered. Since $P_X(w_i, w_j)$ is defined as $\frac{N_{ij}}{N_{**}}$, the proportion of the number of occurences of word pair w_i and w_j to the total number of word pairs in the corpus, the distance function for w_1 and w_2 reduces down to:

$$D(w_1, w_2) = \sum_{1 \le i \le n} \mid N_{w_1 i} - N_{w_2 i} \mid + \mid N_{i w_1} - N_{i w_2} \mid \qquad (8)$$

Having such 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 neighbour 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 that is outside the formed sets, w_i for our case and the process iterates until no word is left not being a member of any set. The formed sets will be the clusters at the bottom of the cluster hierarchy. Then to determine the behaviour of a set, the frequencies of its elements are added and the previous process is carried on the sets this time 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.
- Inadequency of the method used to obtain the set behaviour from its element's properties.

The greedy method results in a nonoptimal clustering in the initial level. To make this point clear consider the following example: Let us assume that four words w_1,w_2, w_3 and w_4 are forming the lexicon space. And 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

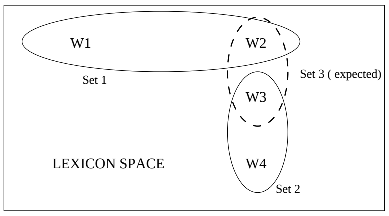

Figure 1: Example for the clustering problem of greedy algorithm in a lexicon space with four different words. Note that d_{w_2,w_3} is the smallest distance in the distribution. However since w_1 is taken into consideration, it forms set1 with its nearest neighbour w_2 and w_3 combines with w_4 and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

the smallest d_{w_1,w_i} for the first word is d_{w_1,w_2}. So the second word will be captured in the set and the algorithm will skip w_2 and continue the clustering process with w_3. At this point, though w_3 is closest to w_2, since it is captured in a set and since w_3 is more 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 $\{w_2, w_3\}$.

The second problem causing unsatisfactory clustering occurs after the initial sets are formed. According to the algorithm after each cluster is formed, the clusters behave exactly like other single words and get into clustering with other clusters or single words. However to continue the process, the bigram statistics of the clusters or in other words the common behaviour of the elements in a cluster should be determined so that the distance between the cluster and other elements in the search space could be calculated. One easy way to determine this behaviour is to find the average of the statistics of all the elements in a cluster. This method has its drawbacks. The points in the search space for the natural language application are very close to each other. Furthermore, if the corpus used for the process is not large, the proximity problem becomes more 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 behaviour 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.

The idea used to find the optimal cluster for each word at the initial step is to form up such initial clusters in the algorithm used in which words are allowed to be members of more than one class. 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 is also formed. As the second step these intersecting sets are combined into a single set. Then the closest two words (according to the distance function) are found in each combined set and these two closest words are taken into consideration as the prototype 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 with 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 with 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 the statistical behaviour of them should be determined, since bigram statistics are used for the process. The statistical behaviour 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 borderlines and even if a word is 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 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 and Geva, 1989) describe such an unsupervised optimal fuzzy clustering. They present the K-means algorithm based on minimization of an objective function. For the pur-

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

pose of this research only the membership function of the presented algorithm 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:

$$u_{ij} = \frac{\mid \frac{1}{d^2(X_i,V_j)} \mid^{\frac{1}{(q-1)}}}{\sum_{k=1}^{K} \mid \frac{1}{d^2(X_i,V_j)} \mid^{\frac{1}{(q-1)}}} \quad (9)$$

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 for all the elements of all classes in the search space are calculated, the bigram statistics of the classes are added up. To find those statistics the following method is used: The bigram statistics of each element is multiplied with the degree of the membership of the element in the working set and 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 behaviour mostly. Hence an element away from a centroid will have a lesser statistical contribution.

4 Results

The algorithm is tested on a corpus 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 and the frequencies of words are collected. Then the most frequent thousand words are chosen and they are sent to the clustering process described in the previous sections. These most frequent thousand words form the 70.4% of the whole corpus. The percentage goes up to about 77% if the next most frequent thousand is added to the lexicon space. The first ten most frequent words in the corpora and their frequencies are presented in Table 1.

The clustering process builds up a tree of words having words on the leaves and clusters on the inner nodes. The starting node denotes the largest class containing all the lexicon space. The number of leaves that is the number of clusters formed at the initial step is 60. The depth of the tree is 8. Leaves appear starting from the $5th$ level and they are mainly located at the $5th$ and $6th$ level. The number of nodes connecting the initial clusters is 18. So on the average about three clusters are combined together in the second step. Table 2 displays two results from the clustering tree. The first one collects a set of nouns from the lexicon space. However the second one is somewhat ill-structured namely two prepositions, two adjectives and a verb cluster are combined into one.

Some linguistic categories inferred by the algorithm are listed below:

- prepositions(1): by with in to and of
- prepositions(2): from on at for
- prepositions(3): must might will should could would may
- determiners(1) : your its our these some this my her all any no
- prepositions(4): between among against through under upon over about
- adjectives(1) : large young small good long
- nouns(1) : spirit body son head power age character death sense part case state
- verbs(1) : exclaimed answered cried **says** knew felt said **or is was** saw did asked gave took made thought **either** told **whether** replied **because though how** repeated open remained lived died lay **does why**
- verbs(2) : shouted wrote showed spoke **makes** dropped struck laid kept held raised led carried sent brought rose drove threw drew shook talked **yourself** listened wished meant **ought seem seems** seemed tried wanted began used continued returned appeared **comes knows** liked loved
- adjectives(2) : sad wonderful special fresh serious particular painful terrible pleasant happy easy hard sweet
- nouns(2) : boys girls gentlemen ladies
- adverbs(1) : scarcely hardly neither probably
- verbs(3) : consider remember forget suppose believe say do think know feel understand
- verbs(4) : keeping carrying putting turning **shut** holding getting hearing knowing finding drawing leaving giving taking making having being seeing doing
- nouns(3) : streets village window evening morning night middle rest end road sun garden table room ground door church world name people city year day time house country way place fact river next earth
- nouns(4) : beauty confidence pleasure interest fortune happiness **tears**

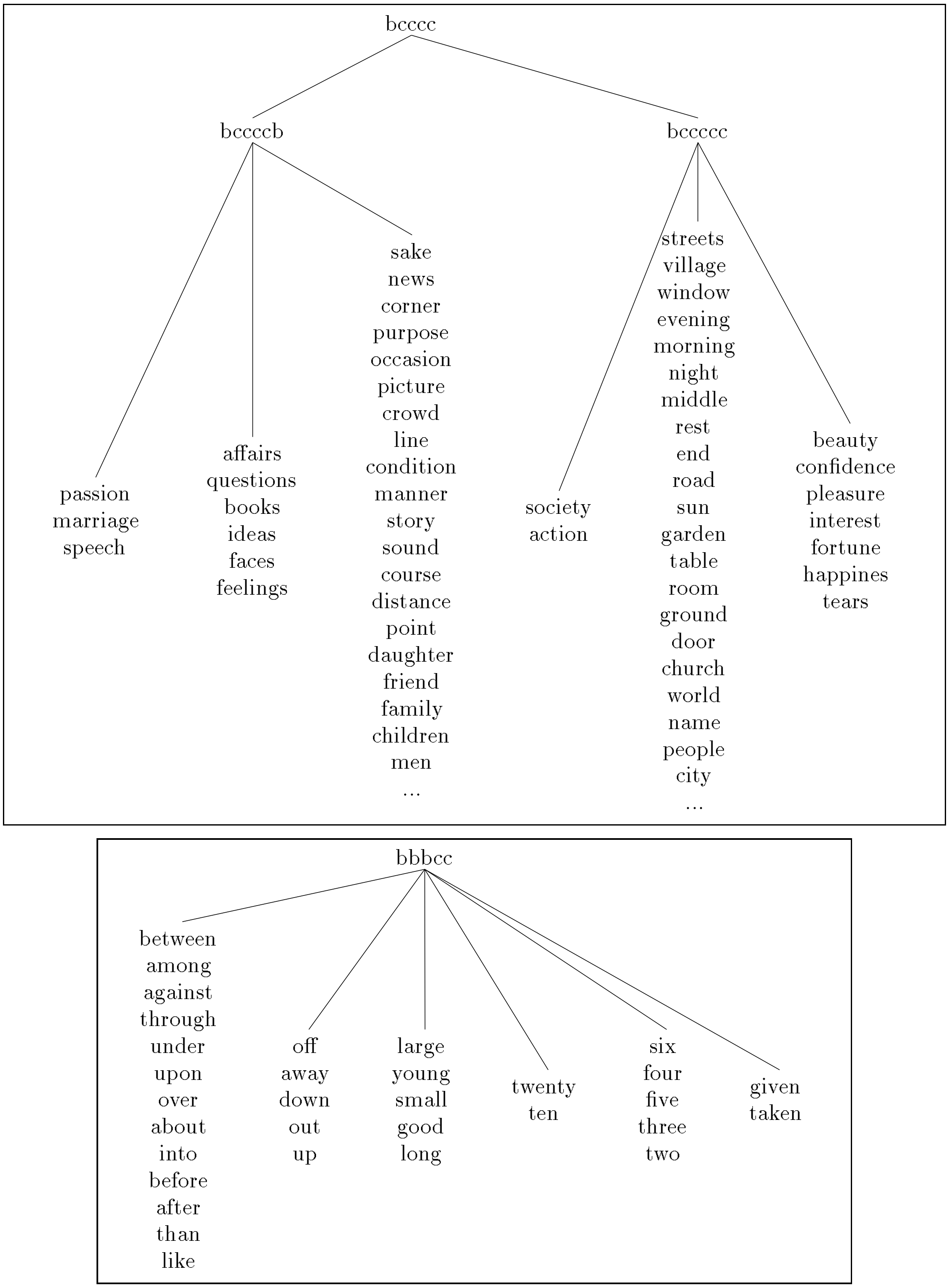

Table 2: Examples from the cluster hierarchy

The ill-placed members in the clusters are shown above using bold font. The clusters represent the linguistic categories with a high success rate ($\approx 91\%$). Some semantic relations in the clusters can also be observed. Group nouns(2) is a good example for such a semantic relation between the words in a cluster.

5 Discussion And Conclusion

It can be claimed that the results obtained in this research are encouraging. Although the corpus used for the clustering is quite small compared to other researches, the clusters formed seem to represent the linguistic categories. It is believed that the incorrect ones are due to the poorness of the knowledge conveyed through the corpus. With a larger training data, an increase in the convergence of frequencies, thus an increase in the quality of clusters is expected. Since the distance function depends on only the difference of the bigram statistics, the running time of the algorithm is quite low compared to algorithms using mutual information. Though the complexity of the two algorithms are the same there is an increase in the efficiency due to the lack of time consuming mathematical operations like division and multiplication needed to calculate the mutual information of the whole corpus.

This research has focussed on adding fuzziness to the categorization process. Therefore different similarity metrics have not been tested for the algorithm. For further research the algorithm could be tested with different distance metrics. The metrics from the statistical theory given in (de Marcken, 1996) could be used to improve the algorithm. Also the algorithm could be used to infer the phrase structure of a natural language. Finch (Finch, 1993) again uses the mutual information to find out such structures. Using fuzzy membership degrees could be another way to repeat the same process. To find out the phrases, most frequent sentence segments of some length could be collected from a corpus. In addition to the frequencies and bigrams of words, the statistics for these frequent segments could be gathered and then they could also be passed to the clustering inference mechanism and the resulting clusters would then be expected to hold such phrases together with the words.

To conclude, it can claimed that automatic word categorization is the initial step for the acquisition of the structure in a natural language and the same method could be used with modifications and improvements to find out more abstract structures in the language and moving this abstraction up to the sentence level succesfuly might make it possible for a computer to acquire the whole grammar of any natural language automatically.

References

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

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

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.

de Marcken, Carl G. Unsupervised Language Acquisition PhD Thesis, 1996.

McMahon, John G. and Francis J. Smith. Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. Computational Linguistics, 1996.

Wilms, Geert Jan. 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.