Distance-based Clustering of words in text Documents

László Bednarik
Department of Information Technology,
Comenius Faculty, Eszterházy Károly College, Hungary

Abstract— In the last decade different applications of computer science (data mining, text mining) have appeared. The task data mining is primarily to reveal structured data in data bases, for example data warehouses. Text mining is assessable information from an unstructured data. At present automatic question generation is a current research area. One of the modules of the framework systems that we worked on does the clustering of words in text documents. This paper shows some implemented clustering methods and some searched algorithms.

Keywords— hierarchic clustering, distance-determination, adaptation of quality threshold algorithm, depth-search, the best first search.

I. INTRODUCTION

The elemental purpose of putting words of text documents into proper clusters is to compose groups from words of a document set in which the occurring words are similar to each other according to some features defined in advance. Properly revealing similarities between words on the base of their content establishes the basis of automatic determination of words from sentences to be extracted for questions, further more alternatives offered to their substitution by algorithms applying knowledge-intensive methods. During clustering words we developed a new conception to define distances. Distances between words are calculated according to the frequency of their common occurrence in sentences of documents. In this way the more sentences containing both words can be found in a document the closer two words are to each other. The concept has been implemented by several different algorithms in order to be able to compare results and to reveal their advantageous or disadvantageous features.

II. AN OVERVIEW OF CLUSTERING

The purpose of clustering is to create distinct groups such that those in one group should be as similar as possible and those in different groups should be different as possible [1].

A formal expression of clustering is as follows (1):

\[
O = \{o\}, \quad d = O \times O \rightarrow \mathbb{R}^+
\]

\[
P_d: \{(O_i)_{i=1}^n\}, \quad O_i \cap O_j = \emptyset, \quad \cup O_i = O
\]

\[
\text{avg} \{d(o_i, o_j) \} \leq \text{avg} \{d(o_i, o_j) \} \quad (1)
\]

where: \(O\) is the set of objects \(o\), \(d\) is the distance function, \(P_d\) is a clustering of \(O\).

The most widely applied standard methods are the hierarchic [2], partitioning [3], hybrid [4], incrementing, or non-incrementing, monothetic versus polythetic [5] and fuzzy [6] methods.

Hierarchic methods were named after for putting elements in a hierarchic data base (in a tree, dendrogram, taxonomy). Data points can be found in leaves of the tree. Each internal point of the tree corresponds to a cluster containing points which can be found under them in the tree. Two main hierarchic processes are distinguished: building from bottom to top and from top to bottom. In accumulating processes building from bottom to top each element is a distinct cluster at the beginning then the process joins clusters being the closest to each other implementing a new cluster one level up in the hierarchy. Demolishing methods building from top to bottom function conversely: they start form one cluster containing all data points and they partition it into smaller clusters then split these on and on. In the case of HAC (Hierarchical Agglomerative Clustering) the two closest clusters are joined together in each step [7]. The condition of a halt is a minimum number of clusters or a maximum distance of contraction.

Steps of the algorithm HAC:
- each element is a distinct cluster,
- determining the two closest clusters,
- joining the two closest clusters into one,
- up-to-date distances,
- continuing the process above as long as the condition for a halt let it do so.

The most well-known hierarchic clustering processes are as follows: based on the smallest, the greatest and an average distance, method Ward, algorithm BIRCH [8] (Balanced Iterative Reducing and Clustering using Hierarchies), an algorithm CURE (Clustering Using Representatives), an algorithm Chameleon.
III. CLUSTERING METHODS AND ALGORITHMS

We implemented a new concept for clustering words of documents (figure 1). The clustering algorithm that was implemented, allows handling documents of several hundreds of pages in practice [9].

A. Distance determination built on the frequency of words occurring together

According to the concept of distance determination built on the frequency of words occurring together, the distance of two words can be defined by the number of sentences in which both words occur together. Distance data determined in this way can be described by a distance matrix. Both rows and columns of the matrix are indexed by words of the document. The value in row \( i \) and column \( j \) is determined by the formula in (2):

\[
S_{ij} = \frac{f_{ij}}{\max(f_i, f_j)} \quad d = 1 - S_{ij}
\]

where: \( S_{ij} \) is a number in the interval \([0, 1]\) representing the distance relation of words \( i \) and \( j \); \( f_{ij} \) is the number of sentences in which both words \( i \) and \( j \) occur; \( f_i \) is the number of sentences in a document in which word \( i \) occur; \( f_j \) is the number of sentences in a document in which word \( j \) occur.

B. Extension of distance calculation to clusters

In the software we developed two task specific improvement of the general method [10] of the algorithm HAC took place. The first method takes advantage of the feature of words to be clustered that their relation can also be described by their relevance to the sentence apart from the distance. This makes them different from points clustered without a deeper link. This method extends the application of formula (2) to clusters instead of words. According to it \( f_i \) means the number of sentences in which all the words belonging to cluster \( i \) occur; \( f_j \) means the number of sentences in which all the words belonging to cluster \( j \) occur; \( f_{ij} \) means the number of sentences in which all the words belonging to both cluster \( i \) and \( j \) occur.

By applying formula (2) consequently each cluster can contain only words occurring at least in one of the sentences together in a document. This concept next to preserving the relevance of words to sentences also restricts the maximum number of words to be clustered. A disadvantage of the method is to recalculate the entire distance matrix after each cluster-contraction step. By extending formula (2) to clusters the distance matrix stores not words but distances of clusters therefore the number and content of clusters change after each contraction. For one step and for the entire steps of the cost function can be defined by (3):

\[
O(N - d), \quad O(N * d * N) = O(N^2 * d)
\]

where: \( O \) is the cost function, \( N \) is the number of sentences to be clustered, \( d \) is the distance of clusters.

In each step of the clustering process clusters whose distance is the smallest according to the formula (2) are contracted. Each case when the distance of a cluster is the smallest from several different clusters ends up in a junction in the search process. Hence there are several solutions for the same task. Finding solutions requires applying tree-traversing algorithms. For increasing the speed of the algorithm clustering was also implemented by a depth limited depth-first-search algorithm next to the traditional breadth-first search algorithm [11].

The target function of the breadth and depth limited depth depth-first-search algorithm is to find the first case in which each word to be clustered in a document is an element of exactly one cluster, moreover no cluster contains two words...
whose distance is exceeds a value given as a parameter. These two implemented algorithms differ from each other in the strategy how they determine the next clustering step after starting from an initial state without clusters. Hence the breadth-first search produces all situations occurred by putting as many words as the number of steps from the initial state, into clusters in each step. Contrarily a depth limited depth-first-search is clustering further the very specific situation as long as a number of steps determined in advance or the entire clustering is done.

Steps of clustering by depth-first-search algorithm are shown in table 1.

<table>
<thead>
<tr>
<th>Number of steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Producing the initial state</td>
</tr>
<tr>
<td>2</td>
<td>Putting the initial state into the list of structures to be analysed</td>
</tr>
<tr>
<td>3</td>
<td>The current substructure should be the last element on the list of structures to be analysed</td>
</tr>
<tr>
<td>4</td>
<td>Coupling clusters of the current substructure in every possible way to determine the distance between the two nearest clusters</td>
</tr>
<tr>
<td>5</td>
<td>If the current substructure does not fulfil the stop condition and the number of substructures on the list of structures is less than the depth-limit of the search then go to step 13.</td>
</tr>
<tr>
<td>6</td>
<td>If the current substructure does not fulfil the stop condition then go to step 9</td>
</tr>
<tr>
<td>7</td>
<td>If the current substructure is not on the list of solutions then put the current substructure on the list of solutions</td>
</tr>
<tr>
<td>8</td>
<td>If the number of substructures on the list of solutions has reached the given maximum number of solutions then go to step 18</td>
</tr>
<tr>
<td>9</td>
<td>Deleting the last substructure being on the list of structures to be analysed from the list</td>
</tr>
<tr>
<td>10</td>
<td>If the list of structures to be analysed does not contain any more substructure then go to step 18</td>
</tr>
<tr>
<td>11</td>
<td>If the final substructure of the list of structures to be analysed does not contain a child substructure on which the terminating search has not been carried out then go to step 9.</td>
</tr>
<tr>
<td>12</td>
<td>The last substructure of the child substructures on the list of structures to be analysed on which the terminating search has not been carried out, is put at the end of the list to be analysed. Go to step 3.</td>
</tr>
<tr>
<td>13</td>
<td>Interpreting clusters whose distance is identical to the distance of the nearest clusters to be compressed after coupling all clusters of the current substructure</td>
</tr>
<tr>
<td>14</td>
<td>Compressing each pair of clusters considered to be compressed whose compression does not upset the compression of other pairs of clusters considered to be compressed</td>
</tr>
<tr>
<td>15</td>
<td>Compressing clusters considered to be compressed in step 14, however, they have not been compressed such that for each pair of clusters considered to be compressed a new substructure is generated in which only the pair of clusters resulting in the generation of the new substructure is compressed. Every new substructure is put on the list of the substructures of the current structure in this step</td>
</tr>
<tr>
<td>16</td>
<td>Putting the first substructure of the current structure at the end of the list of structures to be analysed</td>
</tr>
<tr>
<td>17</td>
<td>Go to step 3</td>
</tr>
<tr>
<td>18</td>
<td>End</td>
</tr>
</tbody>
</table>

C. A task specific adaptation of Quality Threshold algorithm

As the cost of recalculating the distance matrix after each compression of clusters exceeds the possible calculation capacity even if the number of sentences is around a hundred therefore it was necessary to discover an algorithm [12] that carries out calculations for clustering exclusively according to the initial distance matrix. Excluding the continuous recalculation of the distance matrix it is possible to reduce considerably both the calculation time and the storage place required by clustering. However, using only the initial distance matrix obtained at the beginning of clustering, information guaranteeing to put only those words into one cluster which occur at least in one of the sentences together in a document is lost. In breadth and depth search algorithm both determining the distance of clusters and confining the size of clusters were based on this information. According to the rules of the algorithm HAC in an initial state each point (word) is in a distinct cluster therefore it is impossible to know from the distance matrix produced in the initial state in the case of more than two words whether there is a sentence containing them together. In the case when clustering is based on an initial distance matrix in order to confine the size of clusters it is advisable to use a well-known method such as clustering according to the distance between the furthest words of clusters. The implemented algorithm was worked out by adapting this method to algorithm QTC (Quality Threshold Clustering) task specifically by the author. The algorithm defines the diameter of clusters as follows (4):
where: \( d \) is the diameter of the cluster, \( x_i \) is all the objective coordinates of point \( i \), \( x_j \) is all the objective coordinates of point \( j \).

During clustering each cluster is modelled by a disc of a diameter defined in advance. It is similar to clustering based on the distance of the furthest words as the diameter of discs can confine the distance of the furthest points (words) to be put into a cluster. Furthermore this algorithm permits of handling points belonging to several clusters. This extra information can be used to change clustering words belonging to several clusters dynamically and according to current requirements. The target of the function of the optimizing algorithm constructing clusters is to collect all words to be clustered in a document into a minimum number of clusters. Observing the diameter defined uniformly in advance for clusters makes for a limit condition. In the initial state of the clustering process – according to rules of the normal algorithm HAC – each word is put into a separate cluster. During the algorithm the diameter of discs representing clusters are continuously increased hence clusters incorporate words successively. During the procedure if each word of a cluster is an element of other clusters different from the given cluster then the given cluster is deleted.

The algorithm is implemented by the best-first-search strategy [13]. During the procedure the algorithm always increases the diameter of a cluster that results in terminating more clusters after the increase. In order to find it a test is carried out in which states produced by all possible merging of clusters being in the current state of clustering. The condition of merge is that the diameter of clusters should not exceed a formerly given value. States obtained in this way represent the potential subsequent states of the current state. From these states a cluster increasing step resulting in a state containing the least clusters is implemented. In the case of an agglomeration containing several identically minimum numbers of clusters the one found first is implemented. The terminating condition of the procedure is for the diameter of each cluster to be equal to a formerly determined value. As a result each cluster contains one or more words which do not belong to any other cluster; however, they may contain words which belong to other clusters, too. The procedure of clustering with the best-first-search strategy is represented in table 2.

### TABLA 2

<table>
<thead>
<tr>
<th>Number of steps</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Producing the initial state</td>
</tr>
<tr>
<td>2</td>
<td>Analysing the possibility of merging a cluster into all the other clusters</td>
</tr>
<tr>
<td>3</td>
<td>Collecting pairs of clusters considered to be merged in point 2 on a list</td>
</tr>
<tr>
<td>4</td>
<td>If the list does not contain at least one pair of clusters then go to step 7</td>
</tr>
<tr>
<td>5</td>
<td>Stating how many clusters would be terminated after merging them in the case of each pair of clusters on the list</td>
</tr>
<tr>
<td>6</td>
<td>Merging one of the pairs of clusters on the list that results in terminating the greatest number of clusters. In the case of several identical values a pair of clusters found first is merged. Go to step 2</td>
</tr>
<tr>
<td>7</td>
<td>End</td>
</tr>
</tbody>
</table>

The best-first search predicts the cost of reaching the goal from a state \( n \) by a heuristic evaluation function and goes to the state for which this cost is the least. The storage- and time cost of the search is \( O(b^m) \), where \( m \) is the maximum depths of the search space. With a well chosen heuristic function the complexity can be reduced considerably. The rate of reduction depends on the given problem and the quality of the heuristic reduction.

### IV. COMPARING RESULTS OF ALGORITHMS

In the implemented clustering algorithms a potential solution for the problem is each agglomeration in which all words given as an input belong to one or more clusters and none of the clusters can store more words due to the size limit set beforehand. Several solutions may exist for the same clustering problems which differ from each other in their parameters (number of clusters; distance of the furthest words inside a cluster; in a document the number of sentences containing words from clusters appearing together etc.). However the implemented algorithms can find all possible solutions for a problem, in the article – for limiting the content length – only parameters of the first solution were compared. Generalizing the formula \( S_{ij} = f_{ij} / \max(f_{ia}, f_{j}) \) for clusters time of finding a solution depending on the number of words to be clustered is quadratic. The junction index of a tree is put into the index of the value obtained in this way. This number depends on the feature of sentences in a document and is difficult to estimate in advance. Figures 2 show the comparison of run-time request of applied algorithms. Since solutions are a result of several cluster contractions it is clear that in order to find the first solution the breadth-first-search strategy has to go almost all along the whole tree. The diagram clearly shows the extra time requested by the continuous recalculation of the distance matrix contrary to method modelled by discs and implemented by the best-first-search strategy.
V. CONCLUSIONS

Through clustering words in a document based on different conceptions the author determined criteria describing similarity of words. Clustering was carried out on the base of a strategy built on the frequency of words occurring together. Hence the smallest analysable unit, instead of words, is represented by a cluster that means a considerable reduction in resources. The correctness of the developed clustering methods was proven by task specific implementations of breadth, depth, best-first search strategies.

REFERENCES