Hierarchical topic detection in large digital news archives

Exploring a sample based approach

Dolf Trieschnigg
University of Twente
Enschede, The Netherlands
trieschn@cs.utwente.nl

Wessel Kraaij
TNO
P.O. Box 155, 2600 AD Delft, The Netherlands
kraaij@tpd.tno.nl

ABSTRACT

Hierarchical topic detection is a new task in the TDT 2004 evaluation program, which aims to organize a collection of unstructured news data in a directed acyclic graph (DAG) structure, reflecting the topics discussed in the collection, ranging from rather coarse category like nodes to fine singular events. The HTD task poses interesting challenges since its evaluation metric is composed of a travel cost component reflecting the time to find the node of interest starting from the top node and a quality cost component, determined by the quality of the selected node. We present a scalable architecture for HTD and compare several alternative choices for agglomerative clustering and DAG optimization in order to minimize the HTD cost metric. The alternatives are evaluated on the TDT3 and TDT5 test collections.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.5.3 [**Pattern recognition**]: Clustering—*Algorithms,Similarity measures*

Keywords

Information Retrieval, Hierarchical Topic Detection, TDT

1. INTRODUCTION

The Topic Detection and Tracking (*TDT*) project is an annually held evaluation study in the field of TDT organized by the National Institute of Standards and Technology(NIST).

TDT has included a Topic Detection task since its inception in 1996. In this task systems are required to organize news stories in clusters, corresponding to the topics discussed. The result can be regarded as a partition of the corpus, in which each news item is assigned to one and only one partition representing a topic.

The systems are scored by comparing the system result to a manually composed *ground truth*. The cost of a (cluster) structure defines the 'distance' to the ground truth; a better structure has a lower cost. The ground truth is composed by annotators of the Linguistic Data Consortium and consists of manually labelled clusters containing news stories discussing a particular topic. A topic is defined as an event or activity, along with all directly related events and activities. The topics are selected from a random sample of documents from the corpus. The annotation is search guided, i.e. the related stories are found using a search engine. Important to mention is that the annotation for the most recently published TDT 5 corpus is incomplete, that is, there will be no guarantee that every story on each topic will have been located [5]: The search for stories related to one particular topic is ceased after 3 hours, in contrast to previous annotations where the annotators decided when all on-topic stories were found.

The Task Definition and Evaluation Plan of TDT 2004 [6] describes two reasons for introducing a new *Hierarchical Topic Detection* task. The first shortcoming is that a flat partitioned structure does not allow a single news item to belong to multiple topics. Furthermore a flat structure does not allow multiple levels of granularity, i.e. topics cannot be introduced at various levels of detail.

The new HTD task enables stories to be assigned to multiple clusters. Furthermore clusters may be a subset of, or overlap with other clusters. The resulting structure must be characterizable as a DAG with a single root node. The root node represents the complete document collection whereas child clusters further down the DAG represent more specific subsets comprising finer detailed topics. For this initial trial evaluation, the task simplifies treatment of time: the task is treated as retrospective search, i.e. the documents may be processed in any order, in contrast to the old task in which the items should be processed in the order they were published [6].

The metric used for the old Topic Detection task is not suitable for this new task. Allan et al [1] discuss various methods for evaluating hierarchical cluster structures. The TDT 2004 HTD task is evaluated by using the minimal cost method described in Allan et al's paper.

The minimal cost metric finds for each annotated topic the system's optimal cluster, having the lowest cost. This cost consists of a *detection cost* representing the 'goodness' of the cluster and a *travel cost* representing the complicated-

ness to find the cluster. The detection cost is the same as for previous topic detection tasks and consists of a penalty for false alarms and misses, misses have more impact than false alarms however. The travel cost has been introduced to penalize 'powerset' cluster structures, i.e. structures having clusters containing all possible combinations of document sets. The travel cost of a cluster is, independent of its content, related to the shortest path to this cluster from the structure's root cluster. The number of encountered branches and the length of the path are the major components in the travel cost calculation, representing the number of choices a user has to make and the number of cluster titles a user has to read to find the best matching cluster. The score function is parametrized, i.e. the impact of the various cost components is set by using parameters. A more detailed explanation of the metric can be found in Allan et al's paper [1] and the TDT evaluation plan [6].

1.1 Overview of this paper

TNO has participated in the HTD task of TDT 2004. This paper discusses TNO's approach, the experiments on the TDT 3 corpus with this system and the final TDT 2004 results.

We would like to answer the following questions:

- Are conventional agglomerative clustering techniques appropriate for the new HTD task? If not, what additional actions do make these techniques suitable?
- Is the resulting structure intuitive and how does this relate to the minimal cost metric?

Paragraph 2 introduces our approach followed by paragraph 3 outlining related work. Paragraph 4 describes the experiments carried out using this approach. Paragraph 5 discusses the most important results from the experiments and participation in TDT 2004. Conclusions and future work will be outlined in paragraph 6.

2. OUR APPROACH

The corpus for TDT 2004, the TDT 5 test collection, contains news corpora from a number of sources and languages. The total corpus consists of around 400,000 stories (see table 1). The stories are multilingual but all all non-English stories are also available in machine translated English. The system only works with the (translated) English stories. The size of the corpus makes it difficult to use a conventional agglomerative hierarchical clustering approach, like single, complete or average clustering, which uses a distance matrix for building a binary cluster tree. The complexity of such methods usually is $O(n^2 \log(n))$ in time and $O(n^2)$ in space [3]. To illustrate this, a set of 400,000 documents would typically[1] require 80 gigabytes of memory (preferably working memory). After that 80 billion document pair comparisons should be made just to fill the matrix.

The goal of this research is to explore possibilities to make agglomerative clustering scalable for large document datasets.

[1]Using a symmetric distance matrix, $O(\frac{1}{2}n^2)$, optimistically using only 1 byte per comparison

Table 1: TDT 5 corpus statistics

	TDT3	*TDT5*
Arabic stories	0	72,910
English stories	34,600	278,109
Mandarin stories	n.a.	56,486
Total stories	n.a.	407,505
Annotated topics	160	250

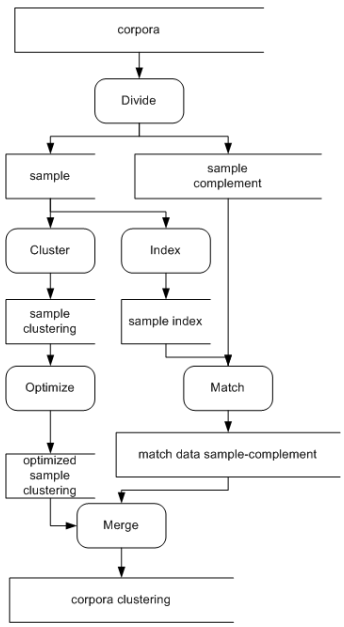


Figure 1: Data Flow Diagram

The following approach has been used:

1. Take a sample from the corpus;
2. Build a hierarchical cluster structure of this sample;
3. Optimize the resulting binary tree for the minimal cost metric;
4. Assign the remaining documents from the corpus to clusters in the structure obtained from the sample.

Figure 1 shows this approach graphically. The steps are explained in the following paragraphs.

2.1 Sampling

The first step is to take a random sample from the corpus. The size of this sample is 20,000 documents, its corresponding distance matrix requires an acceptable 800 megabytes of working memory[2].

2.2 Clustering

The second step is to build a hierarchical cluster structure. Starting point for the clustering method is the cross-entropy reduction scoring function [4]. Suppose we have two documents D_1 and D_2. Both documents are represented by simple unigram language models M_{D_1} and M_{D_2}, a reference

[2]4 bytes per comparison in a symmetric matrix

unigram model for general English M_C is estimated on the complete document collection. Now the cross-entropy reduction (CER) of M_{D_1} and M_{D_2} compared to M_C is defined as:

$$\begin{aligned} CER(D_1; C, D_2) &= H(D_1, C) - H(D_1, D_2) \quad (1) \\ &= \sum_{i=1}^{n} P(\tau_i|M_{D_1}) \log \frac{P(\tau_i|M_{D_2})}{P(\tau_i|M_C)} \end{aligned}$$

where τ_i is an index term and n is the number of unique index terms in C

The generative document model M_{D_2} is smoothed by linear interpolation with the background model M_C [9]. Normalization of scores (by subtracting $H(D_1, C)$) is essential for adequate performance.

The symmetrical version of this scoring function is defined as

$$sim(D_1, D_2) = \frac{CER(D_1; C, D_2) + CER(D_2; C, D_1)}{2} \quad (2)$$

A distance matrix is filled using this scoring function. For the actual clustering 3 basic hierarchic agglomerative clustering methods are used: single, complete and average pairwise linkage.

2.3 Optimizing

The result of this clustering process is a, usually unbalanced, binary tree. An uneven cluster, i.e. a cluster which has childclusters containing an uneven number of documents, adds extra travel cost to all of the clusters below this cluster, especially if this cluster is near the root of the tree. Relating to the real world, the 'user' should consider more branches and titles to find the desired cluster. A more balanced tree will reduce the expected travel cost, but how can the structure be rebalanced without losing clustering information? The metric shows whether the changes to the tree have thrown away clustering information: if after rebalancing the detection cost for any 'optimal' cluster grows, the rebalancing has thrown away valuable information from the original structure. The detection cost should remain the same (or decrease) and the travel cost is decreased.

The method used for rebalancing the tree, without large changes to the optimal clusters, is quite simple. First the clusters are removed which have no documents directly[3] attached and have a dissimilarity higher or equal to a certain threshold. A group of unconnected clusters now remains. These clusters are used to form a better balanced tree with a branching factor of three, suiting the HTD evaluation metric preferring tertiary or quadruple trees[6]. This is done by recursively taking the smallest three (or a different number of) clusters to form a new cluster, until only one root cluster remains.

Figure 2 and 3 show the impact of this rebranching on an average pairwise clustering of 100 documents. The black squares in the bottom of the visualization represent documents, the rectangles represent clusters grouping documents and clusters in new clusters at a higher level. The marked clusters in the first figure will be removed: their dissimilarity is higher than or equal to the chosen threshold and they don't contain documents directly below them. After removing these clusters and corresponding edges, a group of small cluster branches remains. The second figure shows the result of building a more balanced tree with these small branches. The marked clusters now represent newly added clusters.

[3]a directly attached document only appears in this cluster and not in child clusters

2.4 Merging

An index is built from the sample document set. The documents from the corpus which are not in the sample are used as queries on this index returning the best document-likelihood matches. For each document in this complement dataset the best 10 matches are used for merging. The document from the complement dataset is added to all of the matching documents' clusters.

The new documents which don't have any matching documents are collected in one cluster. This ensures new documents at least are assigned to one cluster.

The result of the merging process is a so called fuzzy cluster structure: News items can belong to multiple topics.

3. RELATED WORK

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). It's applied for pattern recognition, image processing, information retrieval and others. The major steps in clustering patterns are: (1) choosing a pattern representation, (2) defining a proximity measure appropriate to the dataset domain, (3) the actual clustering process, (4) data abstraction, e.g. labelling the clusters and optionally (5) assessment of the output. Jain et al give a very good introduction to the concepts of data clustering [3].

In general there are two types of clustering techniques, hierarchical and partitional, which determine the resulting structure. The hierarchical approach produces a nested series of partitions whereas the partitional approach yields a flat structure, in which the relationship between clusters is not as clear as in the first approach.

Clustering techniques can either be agglomerative or divisive. The first is a bottom-up approach which starts with the patterns treated as distinct (singleton) clusters and successively merges clusters together until a stopping criterion is satisfied. Divisive clustering works top-down: the complete dataset is treated as one cluster and splits the clusters until a stopping criterion is met.

A clustering technique can either be hard or fuzzy. Hard clusterings assign each pattern to only one cluster, whereas fuzzy clusterings may assign patterns to multiple clusters based on the degree of membership.

Most hierarchical agglomerative approaches are variants of single link, complete link and minimum-variance (e.g. Ward's method) algorithms. The main difference is the way distance between existing and new clusters is calculated. A very popular partitional method is k-means. By choosing k patterns as initial centroids and assigning the remaining patterns to

Figure 2: Before rebranching, marked clusters will be removed

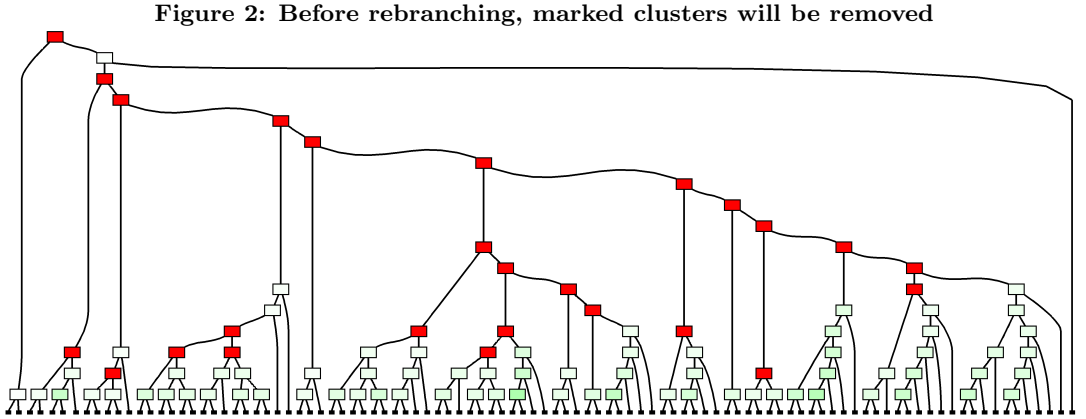
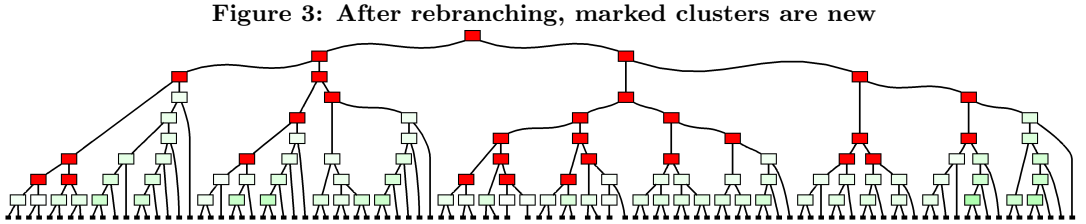


Figure 3: After rebranching, marked clusters are new



one one of these centroids a clustering is obtained. The clusters

Topic detection is a specialization of cluster analysis to facilitate the task of information analysis. Van Rijsbergen [10] formulated the cluster hypothesis for document clustering: "Closely associated documents tend to be relevant to the same requests". Document clustering has been extensively investigated as a methodology for improving document search and retrieval [2].

The high dimensionality of text data and usually large size of datasets do not allow simple application of hierarchical clustering methods because of its high time and space complexity. Much research has been done on finding scalable methods for clustering. This has resulted in different hybrid clustering systems, combining hierarchical and partitional clustering techniques [11, 12].

Cutting et al [2] introduced the Buckshot algorithm, which combines average link clustering with k-means clustering. The average linking is used to find relatively good initial centroids used for further k-means clustering.

Smeaton et al [8] developed a method using a much smaller distance matrix for hierarchical clustering. New documents were added to the clustering by using document-likelihood.

Pantel et al [7] introduced document clustering with committees, which also is a variation on k-means clustering. The centroids are the average feature vectors of carefully chosen committees of patterns representing a cluster.

4. EXPERIMENTS

Experiments were carried out using the English sources from the TDT 3 dataset as a preparation for participation in the trial HTD task of TDT 2004. The size of this dataset is around 35,000 documents, roughly one tenth of the TDT 5 dataset. As a sample we took 10,000 documents from the TDT 3 dataset.

For this sample a symmetric distance matrix was created, filled with the dissimilarity between each document pair. Using this matrix a cluster structure was built using single, complete and average link methods. The sample structure was scored using the minimal cost metric and TDT 3 ground truth containing 160 topics. Based on the bad results for single linkage was decided to exclude this method from further experiments.

Experiments were carried out with rebranching, varying the cut threshold (0, 0.90, 0.95, 0.96, 0.97 and 0.98) and varying the number of branches (3 and 4) to use when glueing the pieces together. Furthermore experiments were carried out applying rebranching before and after the merging process. Other tree simplifying operations were studied, also changing the structure in the lower parts of tree, but these resulted in similar or worse results and are not further discussed in this paper.

The documents in the complement dataset were used as queries for the built sample index. The 20 best matching documents from the sample were searched, using document likelihood. The new documents were assigned to the clusters to which the best matching documents from the sample belong. Two methods were used:

- Adding the new document to the first n (1, 10, 20) matching clusters.

Table 2: Comparison of clustering methods

Method	*Minimum cost*	*Norm. detection cost*	*Norm. travel cost*	*Depth*
Average link	0.2747	0.3722	0.0855	11.68
Complete link	0.6120	0.8778	0.0962	13.14
Single link	0.6970	1.0003	0.1084	24

- Adding the new document to the first matching cluster and to to all matching clusters having a document likelihood higher than a certain threshold (0.5, 1, 2)

The minimal cost was calculated over all of the generated cluster structures, including structures only containing sample documents.

The configuration with optimal result for the TDT 3 test collection was used for the TDT 2004 participation. This was constructing a sample cluster structure using average pairwise link for 20,000 documents, applying a rebranch with branching factor 3 and cut threshold 0.96 and finally merging the best 10 matching documents. Creating the cluster structure of the TDT 5 corpus took around one complete day of processing time on a 900 Mhz machine having 2 Gb of working memory. The sample based clustering system from TNO scored best in the HTD evaluation task of TDT 2004!

5. DISCUSSION

In this paragraph the most interesting results from the experiments and participation in TDT 2004 are discussed.

5.1 Linkage method

First of all the choice of linkage method. The sample TDT 3 dataset was clustered using complete, average and single linkage methods. Average pairwise linking gave the best results by far. For each of the topics in the ground truth, the best cluster, i.e. the cluster having the minimum cost, was calculated. The rows in table 2 show the average characteristics of these best clusters. Without rebranching average pairwise linking gave the best results by far. The metric indicated the cluster structures obtained by using single linkage and complete linkage were much worse.

Further investigation showed that single linkage, as expected[3, 8], performed bad because of its chaining behaviour. A smaller sample of 100 documents was taken, clustered using single linkage and visualized in a tree (figure 4). The figure shows how, especially in the upper part of the tree, new clusters are created by merging an existing cluster and a single document. As a result, the travel cost to reach a more meaningful cluster, i.e. a cluster more closely resembling topics from the ground truth residing at the bottom of the structure, is very high. The travel cost overshadows the detection cost in such a way that the cluster having the lowest overall cost (consisting of travel cost and detection cost) is in the upper part of the structure, although the recall is very poor.

The visualization of a structure with 100 documents obtained by using complete linkage (figure 5) does not clearly show any chaining behaviour. However, details of the experiment outcome showed that a few clusters were chosen

Figure 4: Single link clustering suffers from chaining

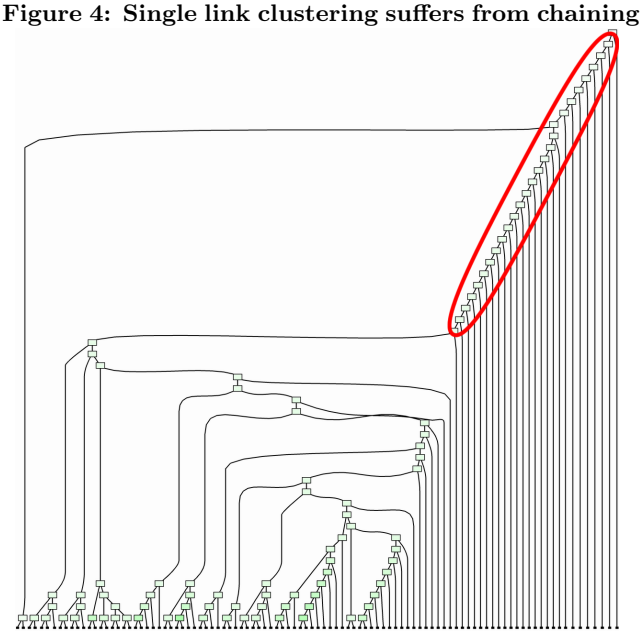
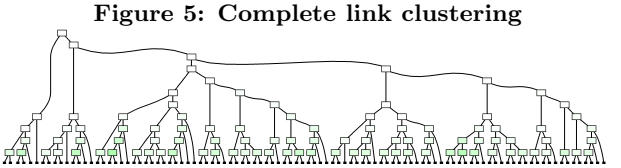


Figure 5: Complete link clustering



frequently as best matching cluster for a topic, just like the single link cluster structure. A screen shot of a cluster structure browser (Figure 6) shows the complete linkage structure also suffers from some kind of 'chaining' behaviour. At the root the document set is divided in two clusters: one tight, relatively small cluster with a dissimilarity little less than 1, and one heavy cluster with a dissimilarity equal to 1. The heavy cluster subsequently is divided again in one small tight cluster and one very heavy cluster. This continues downwards the tree. The visualization of the structure of 100 documents did not show this behaviour, simply because the dataset is too small. The result, just like the single linkage structure, does not allow the best clusters to be found deep down the clustering tree because of the high travel cost to get there. Some of the best matching clusters found (with a smaller travel cost less influencing the complete cost) were promising however. Table 3 gives a sample of the best clusters found for particular topics and its score. The clusters found at depth 2 can be considered as chosen under influence of travel cost - most probably a cluster with a lower detection cost can be found further down the tree. The other clusters however do seem to cover the topics quite well; the recall is quite high, but the precision can be further improved.

The structure obtained by using average linkage seems to be more balanced, naturally enabling more clusters to be considered, not being limited by travel cost. This is one of the major reasons average linkage performs much better when evaluating with the minimal cost metric.

Table 3: Sample of best matching clusters using complete linkage

System cluster	*Minimum cost*	*Norm detect. cost*	*Norm travel cost*	*#Ref*	*#Sys*	*#Union*	*Depth*
v7102	0.6656	1.001	0.0146	5	2	0	2
v7102	0.6656	1.001	0.0146	33	2	0	2
v7102	0.6656	1.001	0.0146	60	2	0	2
v8514	0.2333	0.1686	0.3588	30	29	25	49
v7102	0.6656	1.001	0.0146	1	2	0	2
v7102	0.6656	1.001	0.0146	4	2	0	2
v7102	0.6656	1.001	0.0146	9	2	0	2
v7102	0.6656	1.001	0.0146	1	2	0	2
v8933	0.5553	0.0152	1.6036	10	41	10	219
v7102	0.6656	1.001	0.0146	3	2	0	2
v8500	0.1387	0.0064	0.3954	8	21	8	54
v5701	0.1454	0.0015	0.4247	1	4	1	58
v7102	0.6656	1.001	0.0146	41	2	0	2
v7102	0.6656	1.001	0.0146	5	2	0	2
v7102	0.6656	1.001	0.0146	9	2	0	2
v7102	0.6656	1.001	0.0146	17	2	0	2
v8013	0.2758	0.1765	0.4686	12	30	10	64

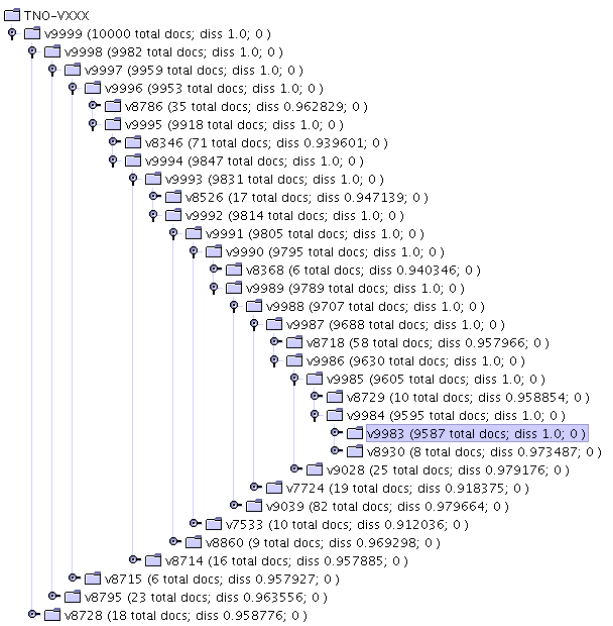


Figure 6: 'Chaining' behaviour of complete linkage clustering

5.2 Influence of rebranching

The preference of the minimal cost metric for clusters closer to the root of the tree in combination with an unbalanced tree resulted in bad results for complete and single linking. The tree should be balanced without modifying important cluster information. This is done using the rebranching method described before. Table 4 shows the minimum cost of the rebranched structures. The dissimilarity thresholds used are adapted to the various methods. The complete linkage causes the dissimilarity to reach 1 quickly for clusters higher in the tree. The threshold is set to 1 correspondingly. Average linkage will not quickly reach a dissimilarity close to 1, so a threshold value smaller than 1 is chosen. Single linkage suffered so badly under the chaining effect, no threshold value could be chosen to build a better tree. Therefore it was decided to not continue further experiments using the single link clustering technique. The rebranching action did not have a big (-6%) impact on the minimum cost for the structure built with average linking. The normalized travel cost however decreased significantly (-65%). As expected the minimum cost for the cluster structure built using complete linking decreased (-50%)as a result of rebranching. A more balanced tree will be searched more thoroughly, i.e. more clusters down the tree are considered, enabling all of the compact clusters to be picked as optimal clusters. As a result the travel cost *and* the detection cost decreased after rebranching the structures built using complete linkage.

5.3 Influence of matching

It was expected that the complete cluster structures, i.e. the structures obtained by adding the rest of the document dataset to the sample structure, would increase the average minimum cost of the optimal clusters. The contrary was true: adding the new documents to multiple clusters, resulting in a fuzzy cluster structure, improved the results! Table 5 shows the cost before and after the matching process. It's particularly interesting that the average detection costs for the TDT 3 and TDT 5 dataset are so different. For the TDT 5 dataset the normalized detection cost and normalized travel cost are in the same order, whereas for the TDT 3 the detection cost is much higher. This might be caused by differences in the dataset, or the way the ground truth was composed.

Furthermore it is noteworthy that the detection cost for the TDT 5 after matching has decreased. The recall has improved (lower P_{miss} rate) but the precision has gone down (higher P_{fa}). The fuzzy matching is causing this. By simply adding new documents to multiple clusters, recall is bound to go up. The cost of a false alarm is very low because the dataset is quite large and topics are quite small[4]. Simply guessing related documents for a particular topic using this

[4]the cost of a false alarm is normalized by the chance a random document does not belong to a topic, which is quite low if the dataset is large and the topics small

matching method pays off: the chance an on-target document is guessed is quite large, resulting in a high chance to increase recall, while the cost of a false alarm is low.

The results give the impression the approach is quite scalable, further investigation has to show this indeed is true and how to explain this performance.

5.4 Intuitiveness

The results do raise questions about the intuitiveness of the metric. For example consider the cluster named 'v18100' in table 6. It represents a topic supposedly to have 81 new items, but itself contains 2826 items, of which 80 actually overlap with the truth cluster. The penalty for missing 1 of the 81 documents is calculated as 0.0123, whereas the false alarm of 2746 items only adds 0.0099 of cost. Just imagine a user trying to find its way through a 'topic' polluted with so many unrelated items. The averages in table 6 show the clusters do have a good recall, but it's precision is terrible. This phenomena also is apparent in the results of table 5: after the fuzzy matching of new documents the the misses decrease but the false alarm rate goes up. The metric allows a large increase of the recall by adding documents to multiple clusters - the loss in precision is not penalized.

Although the idea behind the introduction of the travel cost is intuitive, it does not really penalize 'powerset' structures as it was intended. The travel cost penalizes structures not having the desired branching factor or which are not balanced very well, although the ground truth does not provide any information about this. Another cost component should be introduced to penalize scattering documents over relatively unrelated clusters as is the case when constructing powerset cluster structures.

6. CONCLUSION & FUTURE WORK

In this paper the results of a prototype HTD system were presented. The usage of conventional agglomerative clustering techniques combined with dissimilarity measurement using language modelling looks promising. Cluster structures built with complete linkage using this distance measurement do need restructuring to be effective however. The system has been used for participation in the newly introduced HTD evaluation task of TDT 2004 and achieved best results. The intuitive quality of the clusters is questionable however. At this time the results have too little precision to be really useful. The results give thought about the metric used for HTD evaluation.

Future work should point out what steps in this clustering approach are of most importance and how the precision of the structures can be increased. Furthermore it seems interesting to study the scalability of the system in more depth.

7. REFERENCES

[1] J. Allan, A. Feng, and A. Bolivar. Flexible intrinsic evaluation of hierarchical clustering for TDT. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 263–270. ACM Press, 2003.

[2] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM Press, 1992.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[4] W. Kraaij. *Variations on language modeling for information retrieval*. PhD thesis, University of Twente, May 2004.

[5] Linguistic Data Consortium. TDT 5: Project resources for 2004 evaluation. `http://www.ldc.upenn.edu/Projects/TDT2004`.

[6] NIST. The 2004 Topic Detection and Tracking (TDT2004) task definition and evaluation plan. `http://www.nist.gov/speech/tests/tdt/index.htm`.

[7] P. Pantel and D. Lin. Document clustering with committees. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 199–206. ACM Press, 2002.

[8] A. F. Smeaton, M. Burnett, F. Crimmins, and G. Quinn. An architecture for efficient document clustering and retrieval on a dynamic collection of newspaper texts. In *Proceedings of the 20th BCS-IRSG Annual Colloquium*, 1998.

[9] M. Spitters and W. Kraaij. Unsupervised event clustering in multilingual news streams. *Proceedings of the LREC2002 Workshop on Event Modeling for Multilingual Document Linking*, pages 42–46, 2002.

[10] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.

[11] P. Willett. Recent trends in hierarchic document clustering: a critical review. *Information Processing Management*, 24(5):577–597, 1988.

[12] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM Press, 2002.

Table 4: Influence of rebranching

Cluster method (size)	*Minimum cost*	*Norm. detection cost*	*Norm. travel cost*	*Depth*
Average linkage	0.2747	0.3722	0.0855	11.68
. . . after rebranching (threshold 0.97)	0.2579	0.3620	0.0559	6.11
Complete linkage	0.612	0.8778	0.0962	13.14
. . . after rebranching threshold 1.0)	0.3497	0.5006	0.0567	5.89

Table 5: Influence of matching on average costs

Cluster method (size)	*Minimum cost*	*Norm. detection cost*	*Norm. travel cost*	*P(miss)*	*P(fa)*
TDT3 sample (10,000)	0.2579	0.3620	0.0559	0.3069	0.0112
. . . after matching (35,000)	0.2430	0.3581	0.0195	0.2681	0.0184
TDT5 sample (20,000)	0.0565	0.0629	0.0441	0.0493	0.0028
. . . after matching (278,000)	0.0282	0.0406	0.0041	0.0224	0.0037

Table 6: Sample results from one complete TDT 5 cluster structure

System cluster	*Minimum cost*	*Norm detect. cost*	*Norm travel cost*	*#Ref*	*#Sys*	*#Union*	*P(miss)*	*P(fa)*
v13965	0.0039	0.0045	0.0028	5	261	5	0	0.0009
v15445	0.0023	0.0023	0.0022	1	133	1	0	0.0005
v14140	0.0024	0.0019	0.0035	27	133	27	0	0.0004
v16401	0.0095	0.0131	0.0025	13	759	13	0	0.0027
v18100	0.0411	0.0607	0.0031	81	2826	80	0.0123	0.0099
v3969	0.0013	0.0004	0.0031	1	24	1	0	0.0001
v5859	0.0019	0.0012	0.0032	2	71	2	0	0.0002
v1076	0.0029	0.0018	0.0051	1	104	1	0	0.0004
v3440	0.0019	0.0013	0.0031	2	76	2	0	0.0003
v9072	0.0094	0.0117	0.0050	21	683	21	0	0.0024
v2590	0.0017	0.0005	0.0042	1	28	1	0	0.0001
v8772	0.0448	0.0664	0.0030	63	223	59	0.0635	0.0006
v17828	0.0016	0.0009	0.0030	1	50	1	0	0.0002
v15435	0.0065	0.0073	0.0051	2	417	2	0	0.0015
v17092	0.0037	0.0042	0.0028	5	241	5	0	0.0008
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
average	0.0282	0.0406	0.0041	43.15	1073.1	42.05	0.0224	0.0037