Flexible Intrinsic Evaluation of Hierarchical Clustering for TDT

James Allan, Ao Feng, and Alvaro Bolivar
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003 USA
{allan,aofeng,alvarob}@cs.umass.edu

ABSTRACT

The Topic Detection and Tracking (TDT) evaluation program has included a "cluster detection" task since its inception in 1996. Systems were required to process a stream of broadcast news stories and partition them into non-overlapping clusters. A system's effectiveness was measured by comparing the generated clusters to "truth" clusters created by human annotators. Starting in 2003, TDT is moving to a more realistic model that permits overlapping clusters (stories may be on more than one topic) and encourages the creation of a hierarchy to structure the relationships between clusters (topics). We explore a range of possible evaluation models for this modified TDT clustering task to understand the best approach for mapping between the human-generated "truth" clusters and a much richer hierarchical structure. We demonstrate that some obvious evaluation techniques fail for degenerate cases. For a few others we attempt to develop an intuitive sense of what the evaluation numbers mean. We settle on some approaches that incorporate a strong balance between cluster errors (misses and false alarms) and the distance it takes to travel between stories within the hierarchy.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithms,Measurement

Keywords

Cluster Detection, Hierarchical Clustering, Evaluation

CIKM'03, November 3–8, 2003, New Orleans, Louisiana, USA.

1. INTRODUCTION

Topic Detection and Tracking (TDT) is a research program concerned with organizing a stream of broadcast and print news stories by the events that they discuss [2]. TDT encompasses several tasks, but one of them requires that a system gather arriving news stories into clusters that correspond to real-world events. That task is known in the community as either "cluster detection" or just "detection."

TDT cluster detection requires that a system assign arriving news stories to existing clusters or recognize when a new cluster must be created—i.e., because a new event occurs in the news. To evaluate a system's effectiveness, it is run on an evaluation data set, the generated clusters are compared to reference topics, and a score is generated. The score represents the cost of errors made during clustering [6].

In order to simplify the problem and the evaluation, the cluster detection task made some simplifying assumptions about the nature of news reporting. The community assumed (1) that all news stories are about a single topic—that is, that a news story can be placed in exactly one cluster—and, (2) that any hierarchical relationship between events should be ignored. As a result, systems generate a clustering of the news that is a partition of the data into a non-hierarchical group of topics.

These assumptions are clearly incorrect. We know by observation that some number of news stories cover multiple topics, even if it is a relatively small number. Further, we know that events can be strongly related and that the relationship may be useful to recognize. For example, some events may be supersets of other events (think of a battle within a war or the judicial proceedings within a criminal case). Nonetheless, the simplifying assumptions were useful for getting the research started.

For the TDT 2003 formal evaluation, the community has opted to remove those restrictions. Stories may be assigned to multiple clusters, and systems are expected to generate a hierarchy of stories rather than a flat partitioning.

In this study, we explore the implications this change of task has on evaluating a system's effectiveness. The original evaluation measure is clearly not ideal and must be adjusted. We start in Section 2 by reviewing the TDT cluster detection task in its old and new forms, and by describing the existing measure and why it now fails. In Section 3 we propose several measures and use degenerate cases to discard all but three. We explore those measures in more detail in

Section 4 by considering artificial data to better understand the meaning of a system scores and empirical data on an actual TDT system to see how well existing approaches do. In Section 5 we consider computational complexity concerns and future trends in sparse annotation. We wrap up in Section 6.

It is important to note that this work does not represent official TDT evaluation policy. It is an independent exploration of issues that we hope will be used as fodder for discussion within the community.

2. TDT CLUSTERING

As described above, the purpose of the TDT cluster detection task is to monitor a stream of incoming news stories and organize them by the events that are discussed. The stories are from broadcast as well as newswire sources and come in English, Arabic, or Chinese. The system must not only properly categorize stories into existing clusters, but must also recognize when a new topic appears in the news that requires the creation of a new cluster.

In all cases the system is required to do the cluster detection on the stream as it arrives. That is, decisions about one group of stories must be completed before the next group of stories is presented (groups are the equivalent of about 30 minutes of news). The on-line nature of the task is not important for this study since the evaluation is always done after the entire set of stories has been processed.

2.1 Reference topics

Evaluation is done by comparing system-generated clusters to a "gold standard" generated by the Linguistic Data Consortium [4]. The topics are identified by selecting a random story from the corpus and carrying out "topic development" to find all stories on the corresponding topic. An obvious question that arises is how the LDC determines the scope of a topic—should it include all stories about an election or just a single campaign stump speech? The LDC created "rules of interpretation" that provide it with strict definitions of how to make that decision.

Interestingly, however, the scope of a topic depends upon what story was selected as the seed. If, for example, the story were about a stump speech, that might result in a very limited topic. However, if the story were broader, it the topic might include not just the stump speech but also much more of the campaign.

The upshot of this is that the scope of a reference topic is unpredictable without knowing which story is the seed story. That is, if a story S in topic T is chosen at random and then a new topic T_S is generated using S as a seed, then any of $T = T_S$, $T \subset T_S$, or $T \supset T_S$ might be true.

The ramification of this from an evaluation perspective is that a system cannot possibly be expected to perfectly emulate the reference topics. Any system is likely to be optimized for an average granularity of topic, so will generate some too large and some too small (in comparison to the reference topics). The full impact of this situation was not realized until recently, and it is a major reason that the changes discussed below in Section 2.3 were adopted.

2.2 Current task and evaluation

In the TDT pilot study [1] and in all TDT evaluations through 2002 [5, 11, 12, 13, 14] systems were required to generate a partition of the incoming stories. This meant that each story had to be placed in *exactly* one cluster, even though we know from observation that some stories described multiple events. Also, despite the granularity/scope issue just discussed, systems were required to define strict boundaries on the topics represented by the clusters.

system	relevance judgment	
output	relevant	non-relevant
in cluster	R_+	N_+
not in cluster	R_-	N_-
total	r	n-r

Table 1: Distributions of stories for different judgments.

To calculate the effectiveness of a system, the evaluation software matched each reference cluster with the "best matching" system cluster. When we compare these two types of clusters, we have four different combinations of judgments for each story as shown in Table 1. Here R_+, N_+, R_-, N_- refer to the number of stories in each category respectively. The effectiveness measure P_{miss} (missed detection rate) and P_{fa} (false alarm rate) are defined as:

$$P_{\text{miss}} = \frac{R_-}{r} \tag{1}$$

$$P_{\text{fa}} = \frac{N_+}{n-r} \tag{2}$$

There are two techniques used in TDT to measure the performance of clustering results [6], both are based on detection misses and false alarms. The first, called the cost function, uses a single number to represent the combination of these two kinds of errors. The second, decision error tradeoff (DET) curve, shows the tradeoff between miss detection and false alarm when we change the decision score.

Fiscus and Doddington [6] discuss the cost function used for TDT evaluations. The aim of the TDT cost function is to penalize misses and false alarms. The cost function is defined as a linear combination of P_{miss} and P_{fa} is given by:

$$C_{\text{det}} = C_{\text{miss}} P_{\text{miss}} P(\text{target}) + C_{\text{fa}} P_{\text{fa}} (1 - P(\text{target})) \tag{3}$$

where C_{miss} and C_{fa} are the costs of missed detection and false alarm respectively and $P(\text{target})$ is the prior probability of finding the target (or equivalently what we have called the probability of relevance). Fiscus and Doddington [6] argue that in TDT misses should be penalized much more heavily than false alarms. Hence $C_{\text{miss}} = 10$ and $C_{\text{fa}} = 1$. They fix a constant value for $P(\text{target})$ for all topics; based on corpus statistics they select 0.02 as the constant. The TDT cost function is thus:

$$C_{\text{det}} = 0.2 P_{\text{miss}} + 0.98 P_{\text{fa}} \tag{4}$$

The second evaluation approach is, a DET curve, is generated by sweeping the decision threshold through the score space. At each threshold, we can calculate a missed detection rate and a false alarm rate, and connected points forms the DET curve. Figure 1 gives an example of the DET curve.

In clustering detection, P_{miss} and P_{fa} are calculated by mapping the system-generated clusters to the "truth" clusters. Each "truth" cluster is assigned a system cluster with

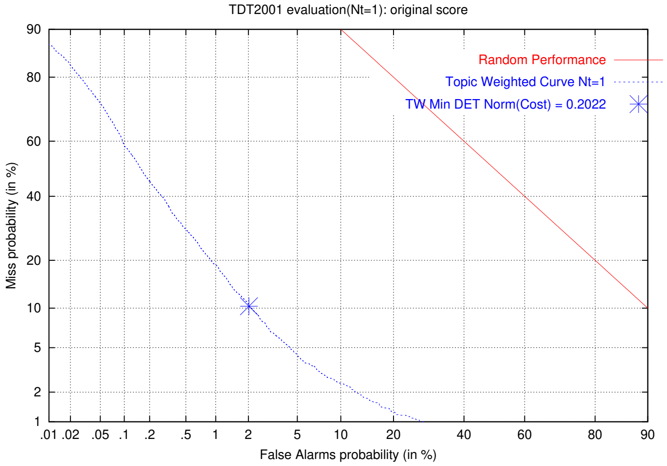

Figure 1: Example DET curve

the smallest detection cost, and it is regarded as the cost for that topic. This mapping produces the global optimal cost.

2.3 New task

For the TDT 2003 evaluation, a system's output permits stories to appear in multiple clusters and encourages a hierarchy to organize the clusters. The purpose of allowing multiple clusters is clear: we know that some stories discuss multiple topics (indeed, the relevance judgments contain instances of stories being judged on-topic for multiple topics).

The change does, however, mean that it is now possible to "game" the evaluation by generating output that is guaranteed to result in perfect (zero) cost. Imagine a system that outputs the power set of the set of stories—i.e., every possible subset is assigned to its own cluster. That means that for any topic (which is just a set of stories) we can find some cluster that contains precisely those stories and no other stories. Since the evaluation model looks for the cluster with minimum cost, it will always find one with zero cost, and the system's output will be deemed "perfect."

Although it is unlikely that any system will generate the power set of a 40,000-story dataset, it seems desirable to change the evaluation measure to prevent this problem from being an issue.

The second change in the evaluation is the hierarchy. This change is a recognition that topic granularity is arbitrary in the evaluation because it depends on the seed story (see Section 2.1). So a system that develops the topic based on a different seed story should not be penalized (much) if it breaks the reference topic into pieces or incorporates the reference topic into a larger topic.

In addition, consider a topic that has 100 stories. A system that breaks the topic into two clusters each of 50 documents will score precisely the same as a system that breaks it into one cluster of 50 and an additional 50 singletons, even though the former is clearly preferable. Because the current evaluation requires a mapping from a topic to a single cluster, we cannot detect this distinction.

As a result, we desire a new measure that incorporates the hierarchy in some meaningful fashion. The intuition is that the hierarchy will be used to pull related clusters together, so that the topic will have different granularity depending where in the hierarchy one stops: one of those should match the reference topic. We see the hierarchy as providing paths for traveling between stories, and we expect that paths between stories in the same topic will be short.

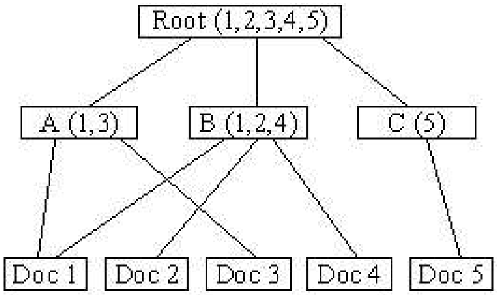

Figure 2: Optimal output for collection 1

Which those issues in mind, we propose some new measures.

3. ALTERNATE CLUSTERING MEASURES

There have been countless attempts to evaluate clustering quality. Many have been in the context of information retrieval where the goal is to improve document ranking using clustering [10, 9, 7, 18, 17]. Others have discussed the use of clustering to impose an order on sets of documents [8, 15, 19, 16]. And, of course, there is work on clustering in the earlier interpretation of TDT clustering [6]. Efforts to evaluate clustering depending on *why* the clustering is being created—i.e., the task determines the best method. Although some existing clustering evaluation approaches could have been tweaked to fit our needs, we found none that seemed to be a satisfactory beginning. We proceed by describing the needs of a clustering evaluation in our context.

Before we define an evaluation measure, we must have a basic mechanism for deciding between useful and useless ideas. What kind of measure is good, what is bad? There are clear answers for flat clustering, but for the new task of hierarchical clustering, there is no known result. Generally speaking, a good evaluation should have low cost (or high value) for known good results, and high cost (or low value) for the results that we do not like. So our first step is to define a collection and its sample results.

To simplify analysis, we use a very small collection that contains just 5 stories: story 1, 2, 3, 4 and 5.

Topic A: story 1, 3
Topic B: story 1, 2, 4
Topic C: story 5

Then we define the optimal output as shown in Figure 2. We should get the best evaluation score (lowest cost) for this graph.

In addition to the optimal output, there are three degenerate cases, all of which should be high cost.

- Degenerate case A: The root node has only one child, which includes all stories (single cluster)

- Degenerate case B: The root has five children, each containing exactly one of the stories, no overlapping (singleton)

- Degenerate case C: The root has $2^5 - 1 = 31$ children, and each of them contains a possible combination of the five stories (power set)

If we use the traditional evaluation method [6] to test degenerate cases C, we will find a cluster that matches perfectly for each topic. It means the evaluation that works well for flat clustering is no longer fit for this application. Tests of other methods get similar results.

We explore many different ways to evaluate the hierarchical structure effectively and efficiently. There are five main models we have tested so far. Some do not work well in degenerate cases, some yield good results for the collections we have used, and some need further research.

3.1 Zero miss, smallest false alarm

For some users, miss detection dominates the cost. They require all the relevant stories to be found but do not care that much about the additional non-relevant ones. And there can be more than one clusters in the hierarchy that have zero miss, so the smallest of them will have fewest false alarm and should be the optimal one.

This algorithm works as following:

```
let Q be a null queue of nodes
insert root into Q
minsize=size(root)
while(Q is not empty)
    take the first element N of Q
    if(size(N)<minsize)
        minsize=$\infty$
    remove N from Q
    foreach child M of N
        if M is a cluster
            calculate M's miss rate
            if $P_{\text{miss}}(M)$=0
                insert M into Q
            fi
        fi
    next
end
```

This measure gets a zero false alarm for the optimal output, but fails for the power set. No matter what the on-topic stories are, it is obvious that a cluster can be found that include exactly these stories. So this degenerate case also gets 0 false alarm and seems as good as the optimal output. It is clear that the power set has exponential size and is very expensive for large collections. It is not a good measure and we will not discuss it further.

3.2 Average distance

In a good hierarchy, we expect to see relevant stories very close to each other. So we can also use the average distance between them as the evaluation measure. Since overlapping is allowed, we may find the same story at different locations. Here we have these agreements:

- If two stories belongs to the same leaf cluster, their distance is 0.

- If two stories do not belong to the same leaf cluster, their distance is the shortest distance between their parent clusters in the hierarchy.

- If a story appears in different locations, the distance between them is not calculated.

- If story A appears in different locations, say A1 and A2, then the distance between other stories and it must be counted twice. So the distances of B - A1 and B - A2 are both included in the sum.

This measure is straight-forward and gets a very small average distance for the optimal output (if we do not consider overlapping stories, the average distance should be 0). But applying it to degenerate case A (single cluster) yields an average distance of 0 because all stories have the same parent! That means the laziest clustering algorithm wins, which is not what we want. So this measure is also dropped.

3.3 Hierarchy traversal

Here we define a cost for the task of finding all stories in a topic. The browsing starts from the root of the hierarchy and searches "optimally" for the on-topic stories. In this case, the cost function is no longer the canonical form, which considers just the miss rate and false alarm rate, but is defined as:

$$Cost = C_{\text{det}} + C_{\text{travel}} \tag{5}$$

Here C_{travel} is the travel cost, which includes two parts:

- Since we must consider all children of a node to find the right link to follow, every branch has a cost (CBRANCH)

- After we choose the right child, following that link also brings some cost (CTITLE—think of it as the cost of reading a title of a story)

This evaluation algorithm runs as following, with "iteration" counting the number of clusters considered while looking for on-topic stories.

```
let Q be a null queue of nodes, sorted by increasing cost
calculate $C_{\text{det}}(root)$
insert root into Q
$C_{\text{travel}}$=0;
iteration=0
while(Q is not empty)
    let N be the first node in Q
    iteration++;
    record iteration and the total cost
    remove N from Q
    $C_{\text{travel}}$+=CTITLE
    foreach child M of N
        if M is a cluster
            $C_{\text{travel}}$+=CBRANCH
            Calculate $C_{\text{det}}(M)$
            Insert M into Q
        fi
    next
end
```

The output of this method is a cost - iteration graph, as shown in Figure 3.

The idea of this measure comes from the DET curve. With a curve instead of a single number, more details of the cost

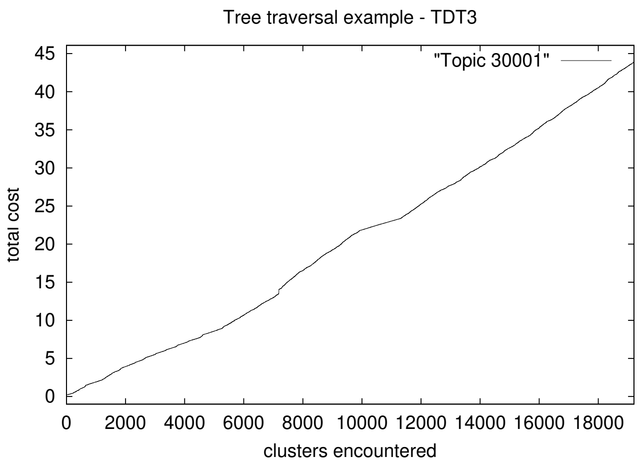

Figure 3: Example cost - iteration graph

change can be found. And we expect to see different shape of curves for different output. To our surprise, all curves for a larger collection (not our two 5-story collection) look like Figure 3, and no useful feature of the curve has been found yet. This measure is put on hold for further analysis and will not appear in later sections.

3.4 Minimal cost

This method is the most similar to the one used in the flat clustering. The only difference is that a travel cost is incorporated into the total cost so as to evaluate the hierarchy. Here the travel cost is defined as the search cost to find the optimal cluster from the root. We use a best-first search to find the optimal node. The number of nodes in the hierarchy is usually large, so we use some trimming algorithm to expedite the search. Once we find some branch that cannot generate any better result than the current optimal value, that branch will be trimmed.

```
let Q be a null queue of nodes, sorted by increasing cost
Ctravel(root)=0
calculate $C_{\text{det}}(root)$
insert root into Q
mincost=$\infty$
while(Q is not empty)
    let N be the first node in Q
    if cost(N)<mincost
        mincost=cost(N)
    remove N from Q
    foreach child M of N
        if M is a cluster
            calculate $C_{\text{det}}(M)$
            $C_{\text{travel}}(M) = C_{\text{travel}}(N) + CBRANCH \times$
                numchild(N)+CTITLE
            if $C_{\text{miss}} P_{\text{miss}}(M) + C_{\text{travel}}(M)$ <mincost
                insert M into Q
            fi
        fi
    next
end
```

When Q is empty, *mincost* is the cost of the optimal node.

Here the total cost is a linear combination of C_{det} and C_{travel}. The weight setting is ad-hoc according to the size of the collection, but biased to C_{det}.

Experiments in collection 1 get good results. The optimal output has a cost much lower than the degenerate cases. More experiments on this measure will be discussed in Section 4.

3.5 Expected travel cost

This evaluation method looks more like a user-oriented measure. Usually a user will not start from the root cluster which contains thousands of stories. Instead, he/she will begin from a single story and try to find all those related to it, and it requires some search algorithm to find all other relevant stories. The frequently-used method is like that:

```
sum=0
foreach on-topic story
    cost=0
    goto(parent)
    cost+=CTITLE
    Find()
    while not all on-topic stories found
        goto(parent)
        cost+=CTITLE
        Find()
    end
    sum+=cost
    next
average cost=sum/numberof(on-topic stories)

Find() {
    foreach child M
        if M is a cluster
            goto(M)
            cost+=CTITLE
            Find()
        else
            cost+=CTITLE
        check story
        fi
    next
}
```

A variation of this method accumulates the cost when each relevant story is found. It is based on the assumption that users do not care about few stories that are scattered in the hierarchy. As long as most relevant stories can easily be found, it is a good result.

Both versions of this measure get the lowest cost for the optimal output, while the non-accumulative version has larger difference. They seem both good measures and require further tests.

4. UNDERSTANDING MEASURES

In section 3, we have dropped two measures using the degenerate cases and one for the obscurity of results. In this section, more experiments will be carried on to compare the rest.

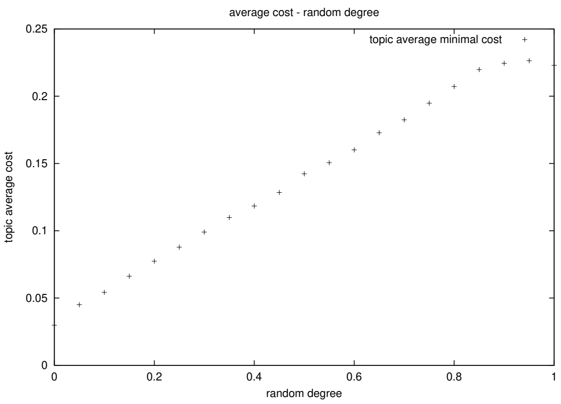

Figure 4: cost - random factor: minimal cost

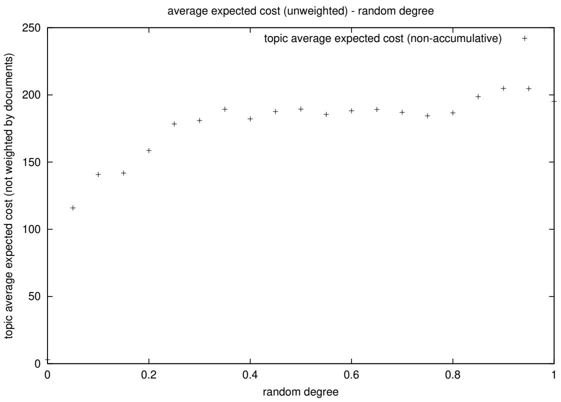

Figure 5: cost - random factor: expected cost (non-accumulative)

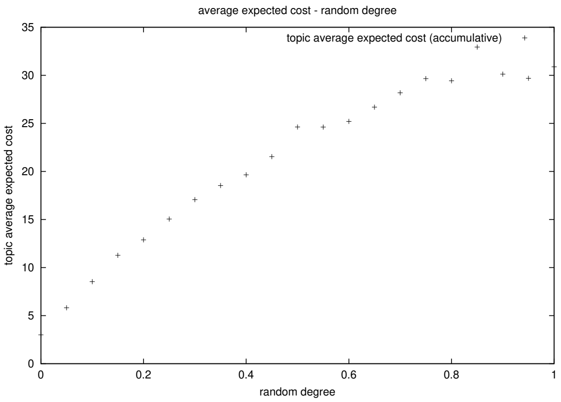

Figure 6: cost - random factor: expected cost (accumulative)

4.1 Comparison Using Standard Data

A good measure must be able to distinguish good results and bad results. It should also be tolerant of small mistakes, e.g., a single story apart from other on-topic stories should be penalized, but only slightly. Here we will observe the change of costs as the optimal output gets worse gradually.

Test collection 2 has 1000 stories and 50 topics, each has 20 on-topic stories. For simplicity, the topics do not overlap each other. Here P(target) is 0.02 for all topics and consistent with Section 2. There are 50 leaf clusters that correspond to the 50 topics. And a parameter - random factor - states the percentage of on-topic stories randomly distributed. If the random factor is 0, each cluster contains 20 stories on the same topic. If it is 1, every story is randomly assigned a cluster. For a random factor of x, $20x$ stories for each topic are randomly distributed, and the rest remain in the cluster. A random factor of 0.5 means every cluster has a half correct and the other half is random. There is another level of clusters between these leaf clusters and the root, and they are randomly generated. Figure 4, 5 and 6 show the change of cost for the three remaining measures along with the change of the random factor.

From these graphs, the non-accumulative expected cost evaluation is more sensitive than the other two. A random factor of 0.05 means only one story of each topic is missed, but the cost increases greatly over perfect clustering. And more misses lead to only slightly larger cost. For an actual system, it is almost impossible to be perfect, so all systems will get similar costs. Such an attribute is not plausible, so we remove this measure.

The other two curves look like linear, which is good for evaluation. When we get a cost, we can simply look up the curve and say, “OK, this system has a cost corresponding to random factor 0.1. Not bad” or “It gets a cost as large as the result of random factor 0.4. A poor result”. Then, no matter how large or small the cost is, we have a way to normalize it to a meaningful score.

4.2 Evaluation on Real Data

Till now we have tested our measures using two collections, but both collections are not real data in TDT, so in this section we need some actual runs to see the performance.

The collection we use is TDT-3, English only, and manual segmentation. This collection has 120 topics, about 40,000 stories and we have complete relevance judgments for each topic. As usual, we must define the optimal output as the baseline. Naturally, the relevant stories for each topic form a cluster. All upper level clusters have a branch factor of 3. Next we will show why 3 is the optimal value of the branch factor using the minimal cost measure.

In this measure, we can always find the optimal node with 0 detection cost since the leaf clusters are “perfect”. So the total cost depends just on the travel cost. In Section 3 it was mentioned that there are two kinds of travel cost: cost to consider a branch and cost to follow a link. Suppose the branch cost is 1 and the link cost is t. Then for a collection with N leaf nodes, if the branch factor is x, then the number of levels is $\log_x N + 1$, the travel cost from the root to a leaf node is

$$(x + t)\log_x N \qquad (6)$$

max-rep	threshold=0.1		threshold=0.2	
	C_{det}	C_{travel}	C_{det}	C_{travel}
1	0.0343	0.0880	0.0372	0.3906
3	0.0232	0.0831	0.0234	0.3776
5	0.0230	0.0880	0.0204	0.3862
10	0.0243	0.0949	0.0122	0.0976

Table 2: Detection cost and travel cost in the minimal cost measure: TDT3, flat clustering

To make it minimal,

$$\frac{\partial(x+t)\log_x N}{\partial x} = 0 \qquad (7)$$

$$x + t = x \ln x \qquad (8)$$

This equation cannot be solved for the general case. But when t=0, it can be solved and x=e. And graphs of Equation 6 shows that 3 is the optimal integer value.

Documents in our test collection were clustered together using an online clustering algorithm: given a stream of documents and a set of clusters, when a new document arrives, a similarity score is calculated for each one of the clusters created so far. The new document is added to the cluster with the highest score that meets a predefined threshold value. A new cluster is created if no such cluster exists. The similarity measure was chosen to be the cosine distance between the new document vector representation and the centroid vector representation for the documents in each cluster. This measure has been one of the most successful up to date for the simpler clustering problem with a flat hierarchy and no overlapping. [3]

In order to allow overlapping and create a hierarchical structure of the data, we implemented minimal variations of the online algorithm described above. Overlapping was allowed by adding a document to the top scoring clusters with a score higher than the threshold value and up to max-rep clusters, where max-rep is a new parameter in the algorithm.

For the hierarchical structure, we decided to take the output from the overlapping clustering algorithm and create a layer of cluster-sets on top of the online clustering system output by clustering together the first round clusters that have at least one document in common. By no means is this the smartest way to solve the clustering problem as defined by the new TDT clustering task, however, for the purpose of understanding our evaluation measures, this output is satisfactory.

Our evaluation results using minimal cost are shown in Table 2. Here all results shown use flat clustering. Comparing to the cost of the optimal output (0.000029, 0.001639), the costs are still large, especially the travel cost. If we adopt a good hierarchical structure, we can expect to see lower travel costs (in a flat clustering the branch factor at the root is immense).

Experiments are also done using the expected travel cost measure. As the evaluation is computationally expensive, we did experiments using only the stories in October (the first one third of the corpus). We take 0.2 threshold, 3, 5 and 10 for max-rep, and the expected costs are listed in Table 3.

The expected travel cost of the optimal output is 9.7261, which is much smaller than those listed in Table 3. The difference comes mainly from the large branch factor in our experiments, and the results will be much better with a well-organized hierarchy.

max-rep	expected travel cost (accumulative)
3	318.9283
5	456.6736
10	521.9299

Table 3: Expected travel cost: TDT3, October stories, 0.2 threshold, hierarchical clustering

When max-rep increases, the detection cost becomes smaller (Table 2) while the expected travel cost gets larger (Table 3). The reason is that they are evaluated in different aspects. The detection cost measures if a cluster is similar to the "ideal" one, so duplications of the same story generates more similar clusters. And the travel cost prefers results where most on-topic stories are near each other, so a duplication causes more outliers. These two measures should have a tradeoff as well as P_{miss} and P_{fa}.

5. OTHER CONSIDERATIONS

The two measures that seem most likely to be useful for evaluating the TDT cluster detection task are minimum cost and accumulative expected travel cost. Our experiments show that the two measure slightly different things and suggest that there may be value in considering their tradeoff. In this section, we consider other aspects of the two measures that trade off against each other.

First, consider computational complexity. We were forced to limit our study of the accumulative expected travel cost to the first third of the test corpus because it was incredibly slow. Suppose we have m topics with n on-topic stories each, then each of the n stories must find $n-1$ stories and has to travel the whole tree in the worst case—even if we do not take overlapping into consideration. The time complexity for it is $O(mnp)$, where p is the number of nodes in the hierarchy.

If we use the minimal cost measure, things are much easier. We can define trimming algorithms to ignore large portions of the tree. Even in the worst case, where no branch can be trimmed, time complexity is just $O(mp)$. Judging from this aspect, the minimal cost measure is way better for a large collection.

On the other hand, trends in test collection annotation suggest that we may not be able to develop complete annotations of all topics. Not only is it expensive to annotate stories for the topics they describe, but the expense will increase greatly if it is necessary to generate a "truth" hierarchy. One model for evaluation that has been proposed is to just annotate randomly selected pairs of stories for whether or not they are on the same topic. This idea could be extended to incorporate a notion of hierarchy also. No topic would be completely annotated and, in fact, most would be only sparsely annotated. How would the proposed measures fare in such a case?

Any measure that depends upon knowing misses and false alarms accurately to just whether a cluster is the best will have troubles. Without knowing what a topic is, it is not possible to find all stories on the topic. On the other hand, measures such as the average distance or expected travel cost can be easily tweaked to handle sparse pairwise judgments. For example, the cost might be the distance between (travel cost) two stories on the same topic, something that should be minimized. And stories on different topics would be expected to have higher travel costs, so if they are too close they would get a higher cost.

At the end of the previous section, we saw that the two favored measures traded off what they showed. We posited that it might be useful to calculate both. It appears that with current assessment approaches, the minimum cost measure is ideal, but that if techniques move toward sparse assessment, expected travel cost is a sounder footing.

6. CONCLUSION

As the TDT cluster detection task moves from a flat partition of the incoming stories to a hierarchical, overlapping clustering, it is necessary for the community to consider alternate evaluation strategies.

In this study we have explored several plausible strategies and settled on two that are reasonable: minimum cost with travel, and accumulative expected travel cost. We believe that the former is preferable at this moment because of the high computational complexity of determining an expected cost.

The TDT research community has been discussing the issue raised here separately. For the sake of easy adoption, it currently appears that the community will select a measure similar to minimal cost, but that penalizes systems directly for generating numerous clusters rather than incorporating travel cost.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[2] James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization.* Kluwer Academic Publishers, Boston, 2002.

[3] James Allan, Victor Lavrenko, and Russell Swan. Explorations within topic tracking and detection. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 197–224. Kluwer Academic Publishers, Boston, 2002.

[4] Christopher Cieri, Stephanie Strassel, David Graff, Nii Martey, Kara Rennert, and Mark Liberman. Corpora for topic detection and tracking. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 33–66. Kluwer Academic Publishers, Boston, 2002.

[5] DARPA, editor. *Proceedings of the DARPA Broadcast news Workshop*, Herndon, Virginia, February 1999.

[6] Jonathan G. Fiscus and George R. Doddington. Topic detection and tracking evaluation overview. In James Allan, editor, *Topic Detection and Tracking: Event-based Information Organization*, pages 17–31. Kluwer Academic Publishers, Boston, 2002.

[7] M.J. McGill G. Salton. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[8] D. Lawrie and W. B. Croft. Discovering and comparing topic hierarchies. In *Proceedings of RIAO 2000 Conference*, pages 314–330, 1999.

[9] A. Leuski. An evaluation of techniques for clustering search results. Technical report, University of Massachuestts, Amherst, MA, 1996.

[10] A. Leuski. Evaluating document clustering for interactive information retrieval. In *Proceedings of the ACM CIKM 2001 Tenth International Conference on Information and Knowledge Management*, pages 33–40, 2001.

[11] NIST. Proceedings of the TDT 1999 workshop. Notebook publication for participants only, March 2000.

[12] NIST. Proceedings of the TDT 2000 workshop. Notebook publication for participants only, November 2000.

[13] NIST. Proceedings of the TDT 2001 workshop. Notebook publication for participants only, November 2001.

[14] NIST. Proceedings of the TDT 2002 workshop. Notebook publication for participants only, November 2002.

[15] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979.

[16] A. Vinokourov and M. Girolami. A probabilistic hierarchical clustering method for organising collections of text documents. Technical Report 5, University of Paisley, Paisley, Scotland, 2000.

[17] E. M. Voorhees. *The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval.* PhD thesis, Cornell University, Ithaca, NY, 1985.

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

[19] Ying Zhao and George Karypis. Evaluation of hierarchical clustering algorithms for document datasets. Technical Report 02-022, University of Minnesota, 2002.