Proceedings of The National Conference
On Undergraduate Research (NCUR) 2005
Virginia Military Institute
Washington and Lee University
Lexington, Virginia
April 21 - 23, 2005

Experiments in First Story Detection

Indro De
Department of Mathematics and Computer Science
Ursinus College
P.O. Box 1000
Collegeville, PA 19426-1000. USA

Faculty Advisor: Dr. April Kontostathis

Abstract

The Linguistic Data Consortium (LDC) has developed data sets of news stories from various sources for the Topic Detection and Tracking (TDT) initiative, an ongoing project for the advancement of text search technologies. A First Story Detection (FSD) application was applied to several data sets from this collection. This task requires identifying those stories within a large set of data that discuss an event that has not already been reported in earlier stories. In this FSD approach, algorithms look for keywords in a news story and compare the story with earlier stories.
Keywords: First Story Detection, TDT, Topic Tracking

1. Introduction

Textual Data Mining (TDM) can be considered a field of its own, containing a number of applications. It has also been also known as text analysis, text mining or knowledge-discovery in text. In general, TDM applications are used to extract non-trivial and useful information from large corpora of text data, which are available in unstructured or structured format. Text mining applications require the use and application of many related fields such as Information Retrieval, Machine Learning, Statistics, and Linguistics. There are various applications of TDM, including in the bioinformatics, market research, consumer trend studies, and scientific research.
We are looking at specific application of TDM, called First Story Detection (FSD) or Novelty Detection. FSD is one of the original tasks of the Topic Detection & Tracking Tasks initiative of the National Institute of Standards and Technology (NIST) [2]. FSD is defined to be the process to find all stories within a corpus of text data that are the first stories describing a certain event [1]. An event is a topic that is described or reported in a number of stories. Examples can be governmental elections, natural disasters, sports events, etc. The First Story Detection process runs sequentially, looking at a time-stamped stream of stories and making the decision based on a comparison of key terms to previous stories.

2. First-Story Detection

In this section, we will discuss the methodology of our research involving First Story Detection (FSD). FSD is closely linked to the Topic Detection task, a process that builds clusters of stories that discuss the same topic area or event [2].

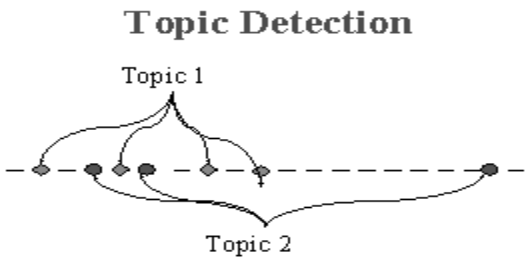

Figure 1: Topic Detection [2]

Comparable to this, FSD evaluates the corpus and finds stories that are discussing a new event. Therefore, FSD is a more specialized version of Topic Detection, because in Topic Detection the system has to determine when a new topic is being discussed. The resulting stories are the "first-stories" we want to retrieve. Figure 2 shows the FSD process.

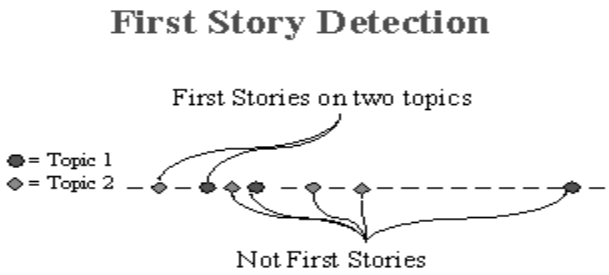

Figure 2: First-Story Detection [2]

A First Story Detection system runs sequentially and creates immediate results while running through the dataset. For the TDT3 corpus, we have a listing of on-topic stories for each of the 120 outlined topics [9]. Using this truth file, we are able to generate a truth file for the FSD task, and directly implement it into our FSD system. Using this method, we are able to generate immediate results and statistical analysis.

3. Previous Approaches

Research in Topic Detection and Tracking started in 1996 with a pilot study (DARPA, University of Massachusetts). In the following, we will discuss several different approaches.

The UMass (University of Massachusetts) approach is based on a clustering-approach of the streaming documents that returns the first document in each cluster as result [6]. Document clusters are groups of documents that appear to be similar in content. Using this approach, and combining it with previously-known solutions to clustering they implemented a modified version of the single-pass (making just one pass through the data) clustering algorithm for the new-story detection. By using a single-pass clustering algorithm, it is possible to run through the stories sequentially, as it is necessary for the FSD task [6].

The UPenn (University of Pennsylvania) approach uses the "single-link" or "nearest-neighbor" technique. This technique stores all stories in clusters of size one, and then merges clusters, if similarities between two clusters are higher than a certain threshold. For the clustering process, a deferral period is defined to be the number of files (including a number of stories) the system is allowed before it relates an event with the stories of that file. An inverted index is then created. After that, all stories are compared to the preceding ones, including stories from a

previous deferral period. When the similarity is high enough, their clusters are merged. If a story can not be combined with any other existing cluster, it becomes a new cluster. These new clusters can be considered new events, thus the story is a first-story [7].

The CMU (Carnegie Mellon University) approach uses the vector-space model to represent each story. It then uses traditional clustering techniques to represent the events. A story is stored as a vector whose dimensions are unique terms from the dataset, and whose elements are the term weights in the story. Term weighting occurs according to simple rules where for example high-frequency terms (which appear in many stories) receive lower weights than terms that seem to have a higher importance in a particular story but not for the rest of the dataset [4, 7]. For the clustering and First-Story Detection, a single-pass algorithm was used.

4. Our Approach

We use a term-weighting system called TF-IDF (Term Frequency x Inverse Document Frequency). Using this technique, each term is assigned with a weight. Our current system uses a two-pass algorithm to assign weights to the extracted terms and store each term and story in a document by term matrix. Several pre-processing steps are performed.

Stop words, words that frequently appear in the English language) are excluded (see example in Table 1). No additional term-filtering or optimization techniques are applied.

Table 1: Excerpt of the stop word list

B	be	became
Because	become	becomes
Becoming	been	before
Beforehand	behind	being
Believe	below	beside

In our FSD system, we look at one source of news stories at a time (for example NBC). The appropriate files are extracted from the whole file list and then combined into a single input file for our program. This input file contains all news stories of the source in a time-sequential order. Additionally, all data is converted to lowercase characters, and special characters are removed to allow for faster and better analysis.

Terms are extracted from the each story (only terms in the title or body of the story were used). Using the number of occurrences of each term within a story and within the corpus, TF-IDF weighting is computed to the term frequency, using a global term-weight. Results are stored in a document by term matrix. A term list is also developed.

$$w_{ij} = tf_{ij} * \log_2 \frac{N}{n}$$

Figure 3: TF-IDF weighting formula [8]

Using the formula from Figure 3, we can assign weights to our terms:
w_{ij} = weight of Term T_j in Document D_i
tf_{ij} = frequency of Term T_j in Document D_i
N = number of Documents in collection
n = number of Documents where term T_j occurs at least once

The algorithm assigns a value to each story (the "FSD-value"). The lower the value, the more likely it is that the story is a first-story. Several methods are used in the acquisition of the FSD-value. As a general rule, the more new high-weighted terms in a story, the more likely it is that the story is a first-story. The system runs and evaluates every story sequentially:

- The first story in a collection is always a first-story (FSD-value 0).

- The second story is evaluated by calculating the occurrences of terms that were in the previous story, thus calculating a measurement of similarity. In our current approach, this story will most likely be a first-story.
- We continue these steps for each subsequent story. If a story contains a high number of previously unknown terms, the FSD-value will be lower. If the FSD-value is under a determined value, the story is identified as a first-story.

Results are stored to a file after the evaluation of each story. After passing all stories in our collection, statistical analysis is performed to measure the performance of our system.

5. Truth Set Evaluation

Unlike in the ETD task, we already have a stable truth set of valid on-topic stories for each of the 120 test topics. Thus it is very easy to produce a specialized truth file for each of the news sources by extracting the appropriate first-stories from the list. However, the news stories contain a lot more events than the 60 test events laid out in the TDT3 study; therefore it is only possible to evaluate our results for first-stories for these events and not for every event. Figure 4 shows the raw truth file including a list of all relevant stories for a certain topic in the collection.

```
<ONTOPIC topicid=30001 level=BRIEF docno=NYT19981228.0443
fileid=19981228_2055_2252_NYT_NYT comments="NO">
<ONTOPIC topicid=30001 level=BRIEF docno=NYT19981229.0004
fileid=19981229_0020_1816_NYT_NYT comments="New in v2.0">
<ONTOPIC topicid=30002 level=YES docno=CNN19981023.0130.0446
fileid=19981023_0130_0200_CNN_HDL comments="NO">
<ONTOPIC topicid=30002 level=YES docno=NYT19981023.0231
fileid=19981023_0041_1801_NYT_NYT comments="NO">
```

Figure 4: TDT3 relevant documents file

In this excerpt, the 3rd document (in bold) is a first-story, as it is the first story that is relevant for topic ID 30002. All following stories associated with this topic ID are still on-topic, but not of importance in our FSD research. Note that the TDT truth file contains all relevant stories, regardless of source. In the example above, the first-story is a CNN document. However, we are running our FSD experiments for one news source at a time, therefore the first-story for a New York Times experiment would be NYT19981023.0231. We first extract all relevant stories of the source we are using for our experimentation from the truth file (for example: CNN, Figure 5).

```
30002  CNN19981023.0130.0446
30002  CNN19981023.1130.0323
30002  CNN19981023.1600.0351
30002  CNN19981023.2130.0265
30002  CNN19981024.1000.0402
30002  CNN19981024.1130.0461
30002  CNN19981024.1300.0378
```

Figure 5: Extracted truth set for the CNN collection

```
30002  CNN19981023.0130.0446
30003  CNN19981017.1000.0129
30004  CNN19981221.1130.0541
30005  CNN19981020.2130.0238
30006  CNN19981005.1600.1233
30007  CNN19981010.1600.0457
```

Figure 6: CNN FSD truth-set extract (only first stories)

This list includes all on-topic documents for each of the 120 standard topics (if such a document exists in this collection). To obtain a list of first-stories, we are extracting all the first stories related to a topic and obtain our final truth-file. We are using this file, to evaluate our algorithm.

6. Results

Because we only have a fixed number of first-stories available, our aim is to produce a system that creates a high recall rate. That is, we want to be able to detect all (or a high number) of first-stories from our truth set.
We use common measurements in information retrieval (P = precision, R = recall).

(1) P = True Positives / (True Positives + False Positives)
(2) R = True Positives / (True Positives + False Negatives)

6.1. MS-NBC test

For this experiment, all MS-NBC stories from the TDT3 collection were used. The data was preprocessed to eliminate special characters, and converted to lowercase. Figure 7 contains information about this collection.

Number of stories:	683
Filesize:	2.04 MB
Lines of data:	37292
Timeframe:	10/01 - 12/31
First stories in truth file:	39

Figure 7: MS-NBC collection overview

The truth set evaluation returned 39 first stories for the MS-NBC data of the TDT3 collection. This means that from the 120 standard topics, 39 were included in this collection. Our application returned 124 first-stories, and was able to detect 13 stories included in the MS-NBC truth set for a recall rate of .33.

30002	MNB19981023.2100.1558	no
30003	MNB19981028.2100.3203	ok
30004	MNB19981218.2100.1558	no
30006	MNB19981013.2100.1687	ok
30012	MNB19981111.2100.2168	no
30015	MNB19981005.2100.2319	ok
30016	MNB19981015.2100.0266	ok
30017	MNB19981120.2100.1804	no

Figure 8: MS-NBC sample results

Figure 9 shows a sample of the results retrieved by our application. The first number in each row represents the topic identification number (e.g. 30002). Because there are 120 standard topics, these topic IDs start with 30001 and go to 30060, then start from 31001 through 31060 (there are two sets of 60 topics each). Looking at the sample above, we can tell that the MS-NBC collection does not contain official first stories for a large number of these standard topics. The 2nd column in each row represents the story ID, including collection name, date, time, and identification number. This ID represents the first story for the appropriate topic. The third column signifies if this story was detected by our FSD system (ok = story detected correctly / no = story not in our results list).

6.2. NBC test

We combined all stories from the NBC (National Broadcasting Company) dataset. Statistics are shown in Figure 9.

Number of stories:	846
Filesize:	1.91 MB
Lines of data:	36953
Timeframe:	10/01 - 12/31
First Stories in truth file:	59

Figure 9: NBC collection overview

The truth set evaluation yielded 59 first stories for the NBC data of the TDT3 collection. Compared to the previous test the NBC collection included a higher number of stories and the truth file had a higher number of first stories, which would improve the significance of our tests. Our application was able to detect 29 stories included in the NBC truth set for a recall rate of .491.

31026	NBC19981017.1830.1375	ok
31031	NBC19981008.1830.0969	ok
31033	NBC19981018.1830.0645	ok
31034	NBC19981108.1830.0898	no
31035	NBC19981003.1830.0062	ok
31036	NBC19981125.1830.0634	ok
31044	NBC19981112.1830.0596	no
31003	NBC19981204.1830.0723	ok
31007	NBC19981021.1830.1238	ok
31008	NBC19981009.1830.1238	ok
31028	NBC19981021.1830.0927	no
31030	NBC19981104.1830.1016	ok
31032	NBC19981012.1830.1224	ok

Figure 10: NBC results sample

Figure 10 shows a particularly good sample of the results retrieved by our application, with 14 out of 20 correctly identified first-stories. Figure 12 shows an extract of the output file created by our system, which included the detected first stories. All retrieved stories from between November 10th and November 25th are shown.

nbc19981110.1830.1244	nbc19981110.1830.1579
nbc19981111.1830.1406	nbc19981112.1830.1285
nbc19981113.1830.0765	nbc19981113.1830.1345
nbc19981113.1830.1565	nbc19981115.1830.0704
nbc19981115.1830.0968	nbc19981115.1830.1156
nbc19981115.1830.1480	nbc19981115.1830.1624
nbc19981116.1830.0758	nbc19981117.1830.0816
nbc19981118.1830.1195	nbc19981119.1830.0738
nbc19981119.1830.1238	nbc19981119.1830.1592
nbc19981120.1830.1367	nbc19981121.1830.1204
nbc19981121.1830.1599	nbc19981123.1830.0485
nbc19981124.1830.1725	nbc19981125.1830.0634

Figure 11: Sample output for NBC test

There are a number of days within this timeframe that have no or just one first-story returned, which is an unusually low number. This is reasonable, as our system takes into account which terms have already occurred in previous stories. The later a story appears in the data set timeframe, the less probable is it that this story is being detected.

However, our system shows some good results for first stories in November and December. In several cases the application retrieved just one single story for a day, but this story was indeed a first-story. A remaining problem was the improvement of the precision of our system. As a general tendency, towards the beginning of the collection, more stories are retrieved. In this time period the application was able to predict almost all the first-stories, however, the precision rate was very low. After a number of processed stories, he number of stories retrieved per day gradually decreases.

It makes sense that more stories will be labeled first stories in the first few days of the system run, as there is no "history" of stories before the beginning of our timeframe (October 1st). However, in the following test, the

threshold was raised as more stories in the database are processed to obtain a higher-quality results set but with an equally strong recall rate.

6.3 ABC test

Figure 12 shows the statistics for the ABC (American Broadcasting Company) collection of the TDT3 data set.

Number of stories:	1481
Filesize:	1.65 MB
Lines of data:	33559
Timeframe:	10/01 - 12/31
first stories in truth file:	61

Figure 12: ABC Collection overview

This test was performed using a modified algorithm, which took into account the location of the story to be processed:

- Our goal is to retrieve fewer stories from the beginning of our dataset and more from the end.
- Each story in the collection is numbered. According to their number, a different rule is applied in order to make the FSD decision.
- Start out with stricter rules to qualify as first story. This means that more new, previously unknown terms are required to qualify as a first story.
- The higher the story number (i.e. the longer the application is running), the fewer new terms are necessary to qualify as a first story.
-

The truth set evaluation yielded 61 first stories, two more than the NBC truth set. Our application was able to detect 24 stories included in the ABC truth set for a recall rate of .393.

30012	ABC19981117.1830.0825	no
30013	ABC19981110.1830.0311	ok
30014	ABC19981018.1830.0414	ok
30015	ABC19981005.1830.0602	no
30016	ABC19981001.1830.0750	ok
30021	ABC19981211.1830.0819	ok

Figure 13: ABC results sample

Figure 13 shows a sample of the results retrieved by our application. The recall rate of .393 was slightly lower than in our NBC test, but still very impressive. Figure 14 shows an extract of the output file created by our system, which includes the detected first stories. This output shows some major differences from the previous test, due to the use of our new algorithm

abc19981223.1830.0221	abc19981223.1830.0438
abc19981223.1830.1042	abc19981223.1830.1559
abc19981224.1830.0314	abc19981224.1830.0709
abc19981226.1830.0161	abc19981226.1830.0708
abc19981226.1830.1424	abc19981226.1830.1537
abc19981227.1830.0397	abc19981227.1830.1057
abc19981227.1830.1074	abc19981228.1830.0000
abc19981228.1830.0322	abc19981228.1830.0342

Figure 14: Sample output for ABC test

The total number of retrieved first stories was only 279, which are 18% of the 1481 total number of stories. Additionally, the stories were a lot more evenly distributed with less first stories in the first part of our collection and more in the later parts (as compared to previous tests).

7. Future Work and Conclusion

Our FSD system was able to detect a relatively high number of the first stories included in our truth files. Previous research in TDT, such as by Makkonen, Ahonene-Myka, and Salmenkivi[10], showed recall rates from 0.190 to 0.294, using their cosine and skew divergence approaches. Our experiments with recall rates up to 0.49 show that our simple approach can be very effective.

Future work must include optimizing the term weighting algorithms and applying them to our current system:

- Elimination of very common terms which are not stop words, but typical to the collection used
- Extraction of noun phrases or n-grams instead of single words
- Algorithms that take into account the total number of stories processed so far (optimization of the technique used in experiment 6.3.)

To improve performance evaluations, it may be necessary to create more standard topics (i.e. more than the 120 topics of TDT3) and test the system on smaller data sets. While our experiments were able to detect a certain number of first stories from the official TDT3 truth sets, it would be interesting to see how many first stories the application was able to retrieve in total.

8. Acknowledgments

I would like to thank my adviser and mentor April Kontostathis for her great support and effort during the last two years. Also, I would like to thank the faculty and staff of Ursinus College to allow and assist me pursuing this research project, including Dean Levy (for providing the funding for the TDT3 data set), Dean Lucas, the Honors committee (Dr. Richard Liston, Dr. Lynne Edwards, and Dr. April Kontostathis), and the Howard Hughes Medical Institute for their generous scholarship enabling me to conduct research during the summer of 2004.

9. References

1. The Linguistic Data Consortium, “The Year 2000 Topic Detection and Tracking TDT2000 Task Definition and Evaluation Plan”, version 1.4, August 2000.
2. National Institute of Standards and Technology, http://www.nist.gov/speech/tests/tdt/
3. James Allan, Victor Lavrenko, Hubert Jin, “First-Story Detection in TDT is Hard”, Center for Intelligent Information Retrieval, University of Massachusetts
4. Salton, G., “Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer”, Reading, MA, 1989.
5. Papka, R., “On-line New Event Detection, Clustering, and Tracking”, Ph. D. Thesis, University of Massachusetts, September 1999.
6. Allan, J., “Incremental Relevance Feedback for Information Filtering”, Proceedings of ACM SIGIR, 1996.
7. Ahmet Vural, "On-Line New Event Detection and Clustering using the concepts of the Cover Coefficient-Based Clustering Methodology”, Masters Thesis, Bilkent University, August 2002
8. Gerard Salton, “Automatic Text Processing”, Chapter 9, Addison-Wesley, 1989.
9. The Linguistic Data Consortium, University of Pennsylvania, May 2004, http://www.ldc.upenn.edu/
10. Makkonen, J., Ahonen-Myka H., Salmenkivi M., “Topic Detection and Tracking with Spatio-Temporal Evidence”, Department of Computer Science, University of Helsinki, Finland