

# TOPIC DETECTION, A NEW APPLICATION FOR LEXICAL CHAINING?

Paula Hatch, Nicola Stokes, Joe Carthy,

Department of Computer Science, University College Dublin,  
Ireland.

{nicola.stokes, paula.hatch, joe.carthy}@ucd.ie

## Abstract

This paper discusses a system for online new event detection as part of the Topic Detection and Tracking (TDT) initiative. Our approach uses a single-pass clustering algorithm, which includes a time-based selection model and a thresholding model. We evaluate two benchmark systems: The first indexes documents by keywords and the second attempts to perform conceptual indexing through the use of the WordNet thesaurus software. We propose a more complex document/cluster representation using lexical chaining. We believe such a representation will improve the overall performance of our system by allowing us to encapsulate the context surrounding a word and to disambiguate its senses.

## 1 Introduction

This paper reports on our work in online new event detection, a new research area initiated by the Topic Detection and Tracking (TDT) project. With the number of online newspapers growing rapidly each year, it is obvious that we are at risk from world news overload, which raises new research challenges for the information retrieval community. TDT concerns detecting the occurrence of a new event such as a plane crash, a murder, a jury trial result, or a political scandal in a stream of news stories from multiple sources and the tracking of a known event. The initiative hopes to provide an alternative to traditional query based retrieval, by providing the user with a set of documents stemming from a generic question like, “What has happened in the news today/ this week/ the past month? ”. This will help the user to lose the feeling of being lost in hyperspace, guiding their browsing by returning sub-clusters of information on the subject of a murder or a bombing for example. If the event interests the user then they can investigate the story further. Computer assisted browsing is at present a popular area in information retrieval from Green’s [1] Hypertext System, to Kazman’s Video Conferencing Indexing System [2].

This research project aims to develop new techniques to attack this problem based on the use of lexical chains and conceptual indexing with the aid of the WordNet computational lexicon. Existing TDT research has focused on the improvement of text-based ranked retrieval techniques previously used for filtering and clustering tasks. We aim to take advantage of these retrieval techniques and to augment them with conceptual indexing. Conceptual indexing requires the identification of the concepts underlying the terms that occur in texts and the construction of a conceptual index of the identified concepts. The conceptual index is made up of lexical chains derived from the documents in the corpus. Lexical chaining has only recently been applied to information retrieval problems with promising results [1-3]. We aim to investigate whether lexical chaining can be used in the area of TDT specifically for new event detection of stories within a news domain.

In the following section, previous work related to event detection and lexical chaining is discussed in more detail. Our system design for new event detection is presented in Section 3, followed by a description in Section 4 of the evaluation methodology used and the results obtained for our two benchmark detection systems. Finally in Section 5 we discuss our future work relating to the incorporation of lexical chaining into our original detection system design.

## 2 Related Work

### 2.1 Background

Topic detection and tracking (TDT) is a DARPA-sponsored project, which investigates the finding and following of new events in a stream of broadcast news stories. The original TDT pilot study had three primary participants, Carnegie Mellon University, Dragon Systems and University of Massachusetts at Amherst, and ran from September 1996 to October 1997. Work on this project is ongoing involving more participants, larger training and test corpora, and a more formally defined meaning of a ‘topic’ called an ‘event’

### 2.2 The Tasks

The goal of TDT is to monitor a stream of broadcast news stories and to find the relationships between these stories based on the real-world proceedings or events that they describe. More specifically, three technical tasks have been outlined within the TDT study [4]:

1. *Segmentation*: Segmenting a stream of data into distinct stories.
2. *Detection*: Identifying those news stories that are the first to discuss a new event occurring in the news. Detection can be further subdivided into online and retrospective detection.
  - *Online Detection*: The system must decide whether or not the current document in the data stream contains a discussion of a new event before looking at the next document in the stream.
  - *Retrospective Detection*: In contrast, the system must decide whether or not a document represents a new event by considering all the documents in the corpus rather than just the documents that came prior to the present document being evaluated.
3. *Tracking*: Given a small number of sample stories about an event, find all the following stories in the stream, about that event.

### 2.3 Online Detection

Our research at the moment is primarily concerned with the detection of events. In particular we consider the CMU [5] and UMASS [6] approaches to online detection. Both groups work on the hypothesis that documents closer together on the input stream are more likely to discuss related events than documents further apart. CMU use the SMART [5] system for document and query representation. Their document clustering approach incorporates either lexical similarity or proximity with a ‘declining influence look back window’, which captures the adaptive nature of a news story with respect to time. UMASS in their approach incorporate this time factor into their ‘thresholding model’, where their matching threshold is continually adjusted to represent the decreasing probability that an event will be reported over time. Recently Papka [7, 13] has furthered the UMASS research effort by experimenting with the larger TDT2 document collection.

In essence all TDT tasks involve the clustering of related documents, for online detection if a document does not belong to any cluster identified up to this point in time then the document is flagged with a YES token indicating a new event otherwise it is flagged with a NO. Each of the TDT participants use a single-pass clustering algorithm in their online detection implementation. Experiments were also completed on an agglomerative hierarchical clustering algorithm [5], which yielded poorer results than the single-pass algorithm. Various cluster comparison strategies for combining similarity results are also applicable and are discussed by Papka in his dissertation [7].

However before a clustering algorithm can be decided upon, a suitable document representation must be chosen. Previous TDT research has tended to focus its efforts on improving classification techniques namely the clustering algorithm used and has ignored the possibility of an alternative to the simple vector space model for document representation. We propose that lexical chains are a suitable alternative for document and event representation, which can be used in conjunction with these new improved clustering strategies.

### 2.4 Lexical Chaining

‘A text or discourse is not just a set of sentences each on some random topic. Rather the sentences and phrases of any sensible text will each tend to be about the same things – that is, the text will have a quality of unity. This is the property of cohesion... it is a way of getting text to hang together as a whole’, Morris and Hirst [8].

A lexical chain is a succession of semantically related words in a text that creates a context and contributes to the continuity of meaning. This concept of representing lexical cohesion using lexical chains was first formulated by Hasan [9, 10], who used them to measure the coherence of stories made up by children. Morris and Hirst then designed an algorithm that automatically built these chains. The lexical chains in a text can be identified using any lexical resource that relates words by their meaning. We use the WordNet thesaurus software [11] as our lexical resource. Morris and Hirsts' original work involved using Roget's International Thesaurus written by Peter Mark Roget in 1852. They were unable to implement their chaining algorithm at the time of writing [8] as a machine readable thesaurus was not available.

Other lexical chaining algorithms have been used in various fields such as hypertext construction [1], multimedia indexing [2, 14 - 16], the detection of malapropisms within text [3] and as a term weighting technique capturing the lexical cohesion in a text [12].

## 3 System Design

### 3.1 The Design of the Benchmark Systems

Our aim in online event detection is to determine as each new document is read in whether or not it concerns a new event. We use a clustering approach to impose organisation on the document collection by grouping together related documents. The general assumption is that mutually similar documents will tend to be about the same topic. A document that concerns a new event cannot be grouped with any previously encountered documents and hence, forms the seed of a new cluster.

In order to cluster documents there are three requirements, a clustering algorithm, a set of clustering metrics and a means of representing a document and a cluster which is conducive to using these.

#### 3.1.1 Document and Cluster Representation

We built two benchmark systems, which differed mainly in the representation of the documents and clusters. In each system, every new document is processed to convert it from its original, raw form to a set of keywords. TDT tags, punctuation, and stopwords are removed. Then, for the first benchmark system called TRAD, words are stemmed and a list of the most important, i.e. the most frequently occurring terms for a particular document, are retained in a term index. For the second benchmark system called SYN, each word is firstly looked up in the WordNet noun file. If found the corresponding synsets (sets of synonymous words) are added to a concept index. Otherwise the word is stemmed and added to a separate term index.

Thus, a traditional vector space model can be used to represent documents and clusters. In the case of the TRAD system a document is represented as a vector, each component of which corresponds to a particular word and whose value reflects the frequency of that word in the document. In the case of the SYN system a document is represented by a pair of vectors, one of which contains synsets and the other any terms not found in the WordNet noun file.

We decided to represent clusters in the same way. Thus, individual documents belonging to a cluster are not preserved. Instead, they are used to build a cluster centroid or prototype, which is a representation of the event, rather than a particular story about the event.

#### 3.1.2 The Clustering Metrics

We chose to use the cosine similarity measure as our clustering metric. The cosine measure computes the cosine of the angle between the two vectors of term weights ( $w$ ) as given in the following formula where the document and cluster vectors are represented by  $d = (w_{1d}, w_{2d}, \dots, w_{td})$  and  $c = (w_{1c}, w_{2c}, \dots, w_{tc})$  respectively:

$$\text{sim}(d, c) = \frac{(d \bullet c)}{(|d| \times |c|)} = \frac{(\sum_{i=1}^t w_{id} \times w_{ic})}{\sqrt{(\sum_{i=1}^t w_{id}^2)} \times \sqrt{(\sum_{i=1}^t w_{ic}^2)}}$$

There are many other types of similarity measure. For example, in [17] a symmetrized form of the Okapi Formula is used to measure the similarity between two documents and in [18] the BBN topic spotting metric, a probabilistic similarity metric derived from Bayes' Rule is discussed.

### 3.1.3 The Clustering Algorithm

For the task of online event detection the entire collection is not available in advance. Each document must be clustered, without deferral, in the order in which it arrives and without access to information relating to subsequent documents in the collection. Thus, we use an incremental approach. An interesting variation of this involves using a look-ahead where a set of documents is processed at once and all clusters are updated simultaneously. An incremental k-means algorithm is described in [18]. There are two processing steps in incremental clustering, selection and thresholding.

#### 3.1.3.1 Selection

The first step involves selecting, according to the clustering metric, the most similar cluster for the document.

- *The Time-Based Selection Model:*

One important aspect of broadcast news is its temporality. When a new event occurs, there are many documents per day that discuss it. However, coverage of an event peaks rapidly and then begins to decline as coverage of newer events replaces it. Eventually, there will be no more stories concerning the original event.

We exploited this knowledge in our selection model by only allowing the  $n$  most recently updated clusters to be compared with the document and hence selected. We thus impose a *time window* within which documents must be clustered. Papka [7] also exploits the temporality of events, but, in a variation on our method, he incorporates the temporal aspect in his thresholding model.

#### 3.1.3.2 Thresholding

The second step of the clustering algorithm involves deciding whether or not the document should be merged with its most similar cluster. If it is merged then the cluster statistics are updated. If it is not merged it is used to seed a new cluster. We chose to use the cosine measure for thresholding as for selection, though it is not necessary to use the same metric for both steps.

- *The Thresholding Model:*

A document is added to its most similar cluster if the similarity between the two exceeds the clusters similarity threshold ( $simt$ ). Thus the similarity threshold of a cluster indicates the level of similarity a document must have with that cluster in order to be judged to be about the same event. If the document is added the cluster centroid is refined and the similarity threshold of the cluster is updated. It becomes the product of:

- the similarity of the *new* cluster ( $c'$ ) and the document just added ( $d$ ).
- a variable parameter called the Cluster Centroid Similarity (CCS) threshold parameter.

Thus,

$$c'.simt = sim(c', d) \times CCS$$

The purpose of the CCS threshold is to indicate the degree of matching with which we are satisfied. Thus, if the threshold parameter is high then the similarity threshold of the cluster will be reduced slowly making it more difficult to add subsequent documents on the stream to that cluster.

## 3.2 The Design of the Lexical Chaining System

Our aim here is to improve the accuracy of the clustering process by using a different representation for a document. We hope that the use of lexical chains, instead of keywords, will result in a more precise characterisation of the topic of a story.

### 3.2.1 Lexical Chain Formation

A lexical chain is a sequence of semantically related words. Such sequences can be identified using a generic algorithm with the following steps:

- Candidate terms are selected from the text.
- An appropriate chain is chosen for each candidate word depending on how the word is related to other words in the chain and the word is added to it.
- Otherwise if no appropriate chain is found then the word is used to start a new chain.

The candidate terms selected are all terms of the text, excluding stopwords. In reality, only nouns, or words which can be morphologically transformed into nouns, can be included in chains. We use only the noun file of WordNet since nouns carry most of the meaning of a text and it is not possible to cross-reference between files in WordNet.

We use WordNet to determine the semantic relatedness between a candidate word and the words of a chain. Each word can be expanded to find its synonyms, hypernyms (is-a), hyponyms (kind-of), holonyms (has-part) and meronyms (part-of) of each of its senses. If an appropriate level of matching is achieved then the candidate term should become part of that chain. St Onge [3] identifies three kinds of relation between words: *extra-strong*, *strong* and *medium-strong* and the permitted paths by which words can be related.

When a new candidate word is added to a chain the ambiguity of both the candidate word and the chain is resolved. The other senses of the candidate word are rejected and the sense of each word in the chain is clarified. Thus, we have automatic sense disambiguation. This is the key to improved classification of stories.

### 3.2.2. Clustering Documents represented by Lexical Chains

We use the same basic incremental clustering algorithm as described in section 3.1 and adapt it to deal with documents represented by sets of lexical chains. We are currently investigating two very different approaches.

The first method involves direct comparison of the lexical chains. Every chain in the incoming document is compared with every chain in each cluster in the permitted time-window. Thus in order to measure the similarity of a document and a cluster we need to be able to:

- Calculate the similarity score between two chains, e.g. a very simple score would correspond to the number of term repetitions and/or the number of terms with shared synsets.
- Calculate the overall similarity score between two sets of chains based on the pair-wise similarities of chains from the two sets, e.g. the sum of the similarity scores of the most similar pairs of chains.

If the similarity score of a document and its most similar cluster does not exceed a particular threshold value then the document is flagged as describing a new event and is used to seed a new cluster. Otherwise, the document should be added to that cluster and the cluster centroid must be updated. This involves merging highly similar chains and eliminating the least dominant chains, thus allowing us to reduce redundancy in the centroid. We use the following features of lexical chains to evaluate chain dominance:

- *Span*: This is the difference in the positions of the first and last words in the chain.
- *Relative Span*: This is the span of a chain divided by the total length of the document in words.
- *Length*: This is the number of words in the chain, including repetitions.
- *Density*: This is the length divided by the span.

The second method involves dealing with document comparisons at a term/synset level rather than at a chain level. We propose that the lexical chains used to represent a document and disambiguate its terms can be further utilised to assign weights to these terms based on the level of importance that a terms chain exhibits within a text. This idea is based on work done by Stairmand [12] who implements this weighting scheme in his QUESCOT IR system. These weights are calculated using a combination of the aforementioned span and density measures of a chain. The advantage of using such an approach is that capturing the degree of similarity between an incoming document and a cluster becomes a trivial matter. Unlike our first method cross chain comparisons are unnecessary as all chains are combined to form a vector of terms and similarities are dealt with at a term level rather than a chain level. In addition when a document representation is added to a cluster centroid representation chain merging is not necessary. Both these processes involving chains are computationally expensive so their simplification should increase the efficiency of the online detection system. We wish to investigate whether or not valuable chain information is lost in this second clustering method causing system effectiveness to suffer as system efficiency increases.

## 4 Evaluation Methodology and Experimental Data

### 4.1 TDT Corpa

The TDT pilot study corpus was created to support research in the areas of segmentation of homogeneous data streams, the detection of new events and the tracking of old or new events within these data streams. The corpus is comprised of both newswire (text) and broadcast (speech) news stories, where each story is represented as a stream of text. It contains nearly 16,000 stories from July 1994 to June 1995 with about half taken from Reuters newswire

and the other half taken from CNN broadcast news transcripts.

The most important and unique element of the corpus is the annotation of the corpus in terms of the events covered in the stories. This annotation data provides a basis for training TDT systems (relevant for tracking) and in particular evaluating performance on a variety of TDT tasks. A set of 25 target events was defined (a subset of the number of events present) which range in diversity from unexpected events like OK city bombing and expected events like Carter visits Bosnia. The stories appear in chronological order and are labeled according to whether a story discusses the event (a YES label), or not (a NO label) and a BRIEF label representing when no more than a tenth of the story discusses the event. For the 25 events of the TDT1 corpus 1132 documents were relevant, 250 documents were judged to contain a brief mention and 10 documents overlapped between the set of relevant documents and the set of brief mentions.

These annotated news articles were collected and created by the Linguistic Data Consortium (LDC). A considerable amount of work went into labeling the stories in the corpus, in order to achieve a convincing labeling approach the corpus was labeled twice by two independent sites. Any differences that emerged from this labeling were reconciled and the resulting labels were hence considered reliable.

A useful addition to the ongoing TDT research effort has been the creation of a second TDT corpus TDT2 containing 60,000 documents. In the initial TDT study, the notion of a topic as explained previously was sharpened to an event where for example the eruption of a Mount Pinatubo in 1991 is an event where as volcanic eruptions in general are seen as a topic. In the TDT2 project the notion of a topic as an event has been broadened, 'A topic is defined to be an event or activity, along with all directly related events and activities'. This means that consequential events such as the funerals of the bomb victims are now considered to be on the same event as the bombing itself. This notion of a topic has yet to be fully resolved however. Also a set of 100 target topics is identified for a mixture of radio TV and newswire sources. Another important component of this TDT2 corpus is that it is divided into three parts, a training set, a development test set (for testing TDT algorithms i.e. diagnostic rather than corpus based training purpose) and an evaluation test set (data reserved for final formal evaluation of performances).

## 4.2 Evaluation Methodology

The purpose of our evaluation is to assess system competence when detecting new events in the document corpus without the use of the relevance judgements used in the TDT evaluation. The TDT evaluation used here is based only on system performance for the 25 events described in the previous section i.e. although many other events or clusters are detected the evaluation is only interested in the 25 reference events. In general, evaluation is described in terms of two different types of error, namely misses (in which the target event is not detected) and false alarms (in which the target event is falsely detected). In addition to these performance metrics, we calculate the traditional recall and precision metrics and the F1 measure. The official evaluation requires that the system output is a declaration (YES flag or NO flag) for each story processed. The evaluation strategy then uses these declarations to calculate the evaluation metrics defined below.

A	The number of new events as defined in the TDT judgement file minus the number of new events missed by the detection system
B	The number of new events falsely detected by the detection system
C	The number of new events missed by the detection system
D	The number of documents that are not about a new event as defined in the TDT judgement file (i.e. total number of documents belonging to the predefined events minus the number of predefined events) minus the number of new events falsely detected by the detection system.

**Table 1: Describes the values used in the calculation of the (online detection system) evaluation metrics, where A, B, C, D are document counts.**

Using Table 1 the evaluation measures are defined as follows:

- Recall =  $r = A/(A + C)$  if  $A + C > 0$ , otherwise undefined.
- Precision =  $p = A/(A + B)$  if  $A + B > 0$ , otherwise undefined.
- Miss =  $1 - \text{Recall} = C/(A + C)$  if  $A + C > 0$ , otherwise undefined.
- False Alarm =  $B/(B + D)$  if  $B + D > 0$  otherwise undefined.

- $F1 = 2rp/(r + p) = 2A / (2A + B + C)$  if  $(2A + B + C) > 0$  otherwise undefined.

Since only 25 events in the corpus were judged, an evaluation methodology developed for the TDT study was used to expand the number of trials. The methodology uses 11 passes through the input stream. The goal of the first pass is to detect the first story on the input stream discussing each of the 25 events. The second pass then excludes these documents, and the goal of the system is then to detect the second document for each of the 25 reference events i.e. the second document artificially becomes the first document for each of the events. This process is then iterated to skip up to ten document in an event. Obviously if an event has less than the required number of documents in order to participate in the iteration, then it is ignored.

### 4.3 Interpretation of Results

In this section we discuss our interpretation of the results obtained from the evaluation of our two benchmark systems mentioned in section 3 of this report. The first system TRAD bases its document representation on stemmed candidate terms extracted from that document, while the second system SYN bases its representations on both weighted terms (possible proper nouns) and weighted synsets taken from the WordNet noun file. The evaluation methodology describe in section 4.2 was used with two purposes in mind,

- To estimate the system parameters.
- To provide a means of comparison with our lexical chaining system results.

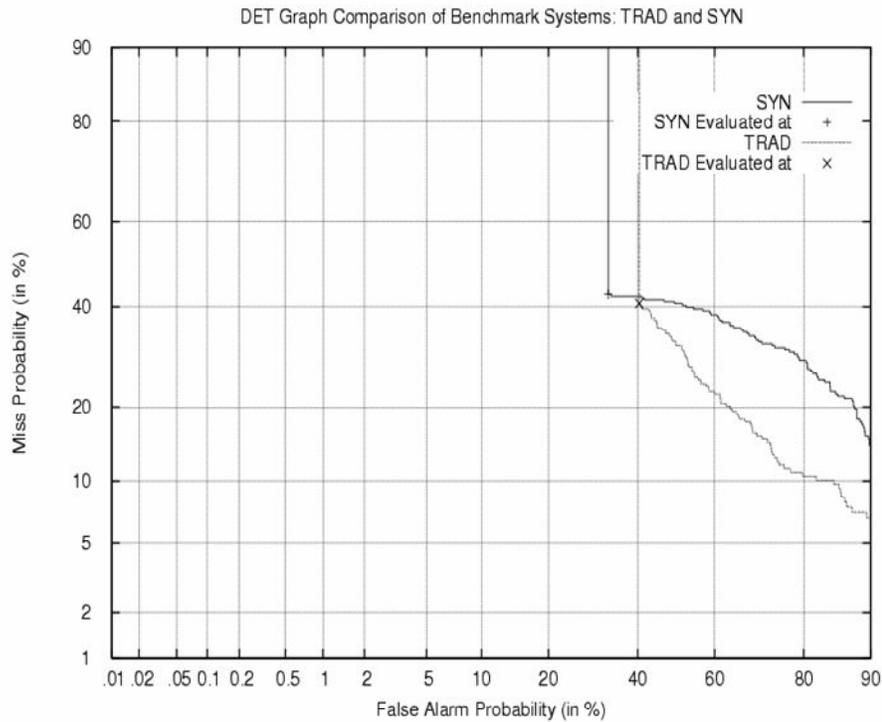
#### 4.3.1 Parameter Estimation

In both benchmark systems three parameters were identified, the *cluster centroid similarity (CCS) threshold parameter*, the *dimensionality parameter*, and the *time window parameter*.

- *The CCS Threshold Parameter*: When a document is added to a cluster, the cluster is given a new similarity threshold based on the similarity of the added document to the new cluster centroid. Since this new threshold value would prohibit the addition of new documents to the cluster (value would be too high to match), it is reduced by multiplying it with the CCS threshold parameter. A number of CCS threshold values (ranging from 0.1 to 0.9) were experimented with varying *dimensionality* and *time window* values. In general it was found that increasing the CCS threshold decreased the precision and miss rate while increasing the recall and false alarm rate. This is explained by the fact that by increasing this threshold only high similarity documents are added to clusters. Thus more documents are identified as discussing new events and so there are fewer misses and more false alarms.
- *The Dimensionality Parameter*: The *dimensionality parameter* relates to the number of terms/synsets used to represent a document and similarly a cluster centroid. Two sets of experiments were carried out. Firstly the number of terms used in the centroid was varied while the number of terms used to represent a document was kept static. Little or no change in the evaluation metrics was seen in this experiment. In the second experiment the number of terms used to represent both a document and a centroid were varied. In this case precision and miss rates increased as the dimensionality increased.
- *The Time Window Parameter*: The *time window parameter*  $n$  determines how many of last  $n$  created clusters a document will be compared with. This idea is based on the news story proximity characteristic, which states that documents in close proximity to each other on the input stream tend to be about the same event. The inclusion of this parameter also considerably improves the run time efficiency of the benchmark systems. Interestingly, we found that a look back threshold of 30 clusters gave similar results to those obtained when the *time window* was increased to the total number of clusters created. This is explained by the fact that the events presented in the judgement file show that most documents within an event are no more than 30 documents apart on the input stream.

#### 4.3.2 Benchmark Systems Performance

The following detection error tradeoff (DET) graph illustrates the tradeoff between misses and false alarms when evaluating our detection systems SYN and TRAD. The points on a DET graph are determined from the false alarm and miss rates belonging to the threshold parameters with the 'best' over all evaluation metrics. The graph is analogous to a recall-precision graph, which is used to depict the tradeoffs between recall and precision values. The decision scores for each document over the 11 iterations are pooled and sorted by score, so that the miss and false alarm rates can be plotted in that order. Both the SYN and TRAD systems are evaluated based on this graph, where points closer to the origin indicate better overall performance. The graph also contains two evaluation points indicating the pool average performance values of the two systems.



**Figure 1: Comparison of Benchmark systems SYN and TRAD.**

System	Miss Rate	F/A Rate	Recall	Precision	F1
TRAD	41%	40.41%	59%	3%	0.06
SYN	43%	32.73%	57%	4%	0.07

**Table 2: Pooled average performance values for the benchmark systems.**

Both Figure 1 and Table 2 show that the TRAD system has miss rates that are lower than the SYN system between false alarm rates of about 35% and 90%. On average however the TRAD system detected only one more event than the SYN system. On the other hand the SYN system shows the lowest overall (pooled) average false alarm rate of 32.73% a difference of almost 8%. Similar results are observed for recall and precision values, where the overall F1 measure between the two systems differs by only 0.01. Thus based on the DET graph and pooled average values we can conclude that no significant difference in performance exists between our two benchmarks systems.

## 5. Conclusions and Future Work

We have designed and implemented two benchmark systems. TRAD demonstrates the use of traditional keyword IR techniques and SYN uses keyword techniques augmented with WordNet derived synsets. As expected the results from these systems are not particularly good. This can be attributed to the following features of the systems.

- In TRAD only *exact* word matches are considered. Thus, an article that discusses “cars” would not match with an article about “automobiles” even though the meaning of the two words is the same. This is because keyword matching rather than *concept* matching is used.
- In SYN an attempt was made to incorporate concept matching by using synset identifiers rather than specific words to represent a document. However, the problem here is that many words have more than one sense so when a word is added to the document’s concept index, other irrelevant senses of the word will also be added. A method of *sense disambiguation* is required to prevent this.
- The clustering algorithm we have used is not as sophisticated as those used by [4 - 7].

We intend to proceed in future work by implementing a lexical chaining system. In this system a document is represented by a set of lexical chains, rather than by a “bag of words”. We expect that there will be a significant improvement in the results obtained. Concepts will be represented through the use of WordNet synsets and sense disambiguation will be accomplished automatically as the chains are constructed. We intend to use the same simple clustering technique but if we achieve an improvement for this algorithm then we would expect to achieve a similar level of improvement for a more sophisticated clustering approach [4 - 7].

## 6. References

- [1] Stephen J Green, *Automatically Generating Hypertext By Comparing Semantic Similarity*, University of Toronto, Technical Report number 366, October 1997.
- [2] Rick Kazman, Reem Al-Halimi, William Hunt, Marilyn Manti, *Four Paradigms for Indexing Video Conferences*, IEEE Multimedia Vol. 3, No. 1, Spring 1996.
- [3] D. St-Onge, *Detection and Correcting Malapropisms with Lexical Chains*, Dept. of Computer Science, University of Toronto, M.Sc Thesis, March 1995.
- [4] James Allan et al., *Topic Detection and Tracking Pilot Study Final Report*, In the proceedings of the DARPA Broadcasting News Transcript and Understanding Workshop 1998, pp. 194-218.
- [5] Yiming Yang, Tom Pierce, Jamie Carbonell, *A study on Retrospective and On-Line Event Detection*, Carnegie Mellon University, In the proceedings of SIGIR 1998.
- [6] James Allan, Ron Papka, Victor Lavrenko, *On-line New Event Detection and Tracking*, University of Massachusetts, Amherst, In the proceedings of SIGIR’98, pp. 37-45.
- [7] Ron Papka, *On-line New Event Detection, Clustering and Tracking*, Department of Computer Science, UMASS, Amherst, PhD Dissertation, 1999.
- [8] Jane Morris, Graeme Hirst, *Lexical Cohesion by Thesaural Relations as an Indicator of the Structure of Text*, Computational Linguistics 17(1), March 1991.
- [9] R Hasan, *Coherence and Cohesive Harmony*, in J.Flood (ed), *Understanding Reading Comprehension*, IRA: Newark, Delaware, 1984.
- [10] M Halliday, R Hasan, *Cohesion in English*, Longman: 1976.
- [11] George Miller, Special Issue, *WordNet: An On-line Lexical Database*, International Journal of Lexicography, 3(4), 1990.
- [12] Mark A. Stairmand, William J. Black, *Conceptual and Contextual Indexing using WordNet-derived Lexical Chains*, Proceedings of BCS IRSG Colloquium 1997, pp. 47-65.
- [13] Ron Papka, James Allan, Victor Lavrenko, *UMASS Approaches to Detection and Tracking at TDT2*, Center for Intelligent Information Retrieval, Computer Science Department, UMASS, Amherst, 1999.

- [14] Rick Kazman, John Kominek, *Supporting the Retrieval Process in Multimedia Information Systems*, Proceedings of HICSS '97, Vol. VI, pp. 229-238.
- [15] Reem Al-Halimi, Rick Kazman, *Temporal Indexing through Lexical Chaining*, in *WordNet: An Electronic Lexical Database and Some of its Applications*, C. Fellbaum (ed.), MIT Press, 1997.
- [16] Rick Kazman, John Kominek, *Accessing Multimedia through Concept Clustering*, Proceedings of CHI '97, March 1997, pp. 19-26.
- [17] S. Dharanipragada, M.Franz, J.S. McCarley, S. Roukos, T. Ward, *Story Segmentation and Topic Detection for Recognised Speech*, IBM T.J. Watson Research Centre, Eurospeech 1999.
- [18] Frederick Walls, Hubert Jin, Screenivasa Sista, Richard Schwartz, *Topic Detection in Broadcast News*, BBN Technologies, Eurospeech 1999.