First Story Detection using a Composite Document Representation.

Nicola Stokes, Joe Carthy,
Department of Computer Science,
University College Dublin,
Ireland.

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

ABSTRACT

In this paper, we explore the effects of data fusion on First Story Detection [1] in a broadcast news domain. The data fusion element of this experiment involves the combination of evidence derived from two distinct representations of document content in a single cluster run. Our composite document representation consists of a concept representation (based on the lexical chains derived from a text) and free text representation (using traditional keyword index terms). Using the TDT1 evaluation methodology we evaluate a number of document representation strategies and propose reasons why our data fusion experiment shows performance improvements in the TDT domain.

Keywords

Lexical Chaining, Data Fusion, First Story Detection.

1. INTRODUCTION

The goal of TDT is to monitor and reorganize a stream of broadcast news stories in such a way as to help a user recognize and explore different news events that have occurred in the data set. First story detection (or online new event detection [1]) is one aspect of the detection problem which constitutes one of the three technical tasks defined by the TDT initiative (the other two being segmentation and tracking). Given a stream of news stories arriving in chronological order, a detection system must group or cluster articles that discuss distinct news events in the data stream. The TDT initiative has further clarified the notion of topic detection by differentiating between classification in a retrospective (Event Clustering) and an online environment (First Story Detection). In FSD the system must identify all stories in the data stream that discuss novel news events. This classification decision is made by considering only those documents that have arrived prior to the current document being evaluated, forcing the system to adhere to the temporal constraints of a real-time news stream. In other words the system must make an irrevocable classification decision (i.e. either the document discusses a *new event* or *previously detected event*) as soon as the document arrives on the input stream. The goal of event clustering on the other hand is to partition the data stream into clusters of related documents that discuss distinct events. This decision can be made after the system has considered all the stories in the input stream.

In addition to defining three research problems associated with broadcast news, the TDT initiative also attempted to formally define an event with respect to how it differs from the traditional IR notion of a subject or a topic as defined by the TREC community. An *event* is defined as 'something that happens at some specific time and place (e.g. an assassination attempt, or a volcanic eruption in Greece)'. A *topic* on the other hand is a 'seminal event or activity along with all directly related events and activities (e.g. an investigation or a political campaign)' [1]. Initial TDT research into event tracking and detection focused on developing a classification algorithm to address this subtle distinction between an event and a topic. For example successful attempts were made to address the temporal nature of news stories[1] by exploiting the time between stories when determining their similarity in the detection process [1]. However current research is now focusing on the use of NLP techniques such as language modeling [2, 3], or other forms of feature selection like the identification of events based on the domain dependencies between words [4], or the extraction of certain word classes from stories i.e. noun phrases, noun phrases heads [5]. All these techniques offer a means of determining the most informative features about an event as opposed to classifying documents based on all the words in the document. The aim of our research is also based on this notion of feature selection. In this paper we investigate if the use of lexical chains to classify documents can better encapsulate this notion of an event. In particular we look at the effect on FSD when a composite document representation (using a lexical chain representation and free text representation) is used to represent events in the TDT domain.

[1] Stories closer together on the input stream are more likely to discuss the same event than stories further apart on this stream.

In sections 2 and 3 we describe the first component of our composite document representation derived from lexical chains, with a subsequent description of FSD classification based on our data fusion strategy in Section 4. The remaining sections of this paper give a detailed account of our experimental results, concluding with a discussion of their significance in terms of two general criteria for successful data fusion.

2. LEXICAL CHAINING

A lexical chain is a set of semantically related words in a text. For example in a document concerning cars a typical chain might consist of the following words {vehicle, engine, wheel, car, automobile, steering wheel}, where each word in the chain is directly or indirectly related to another word by a semantic relationship such as *holonymy*, *hyponymy*, *meronymy* and *hypernymy*.

When reading any text it is obvious that it is not merely made up of a set of unrelated sentences, but that these sentences are in fact connected to each other in one of two ways cohesion and coherence. As Morris and Hirst [6] point out cohesion relates to the fact that the elements of a text 'tend to hang together'. Whilst coherence refers to the fact that 'there is sense in the text'. Obviously coherence is a semantic relationship and needs computationally expensive processing for identification, however cohesion is a surface relationship and is hence more accessible. As indicated by Halliday and Hasan [7] cohesion can be roughly classified into three distinct classes, *reference*, *conjunction* and *lexical cohesion*. Conjunction is the only class, which explicitly shows the relationship between two sentences, *'I have a cat and his name is Felix'*. Reference and lexical cohesion on the other hand indicate sentence relationships in terms of two semantically same or related words. In the case of reference, pronouns are the most likely means of conveying referential meaning. For example in the following sentences, ' *"Get inside now!" shouted the teacher. When nobody moved, he was furious*'. In order for the reader to understand that 'the teacher' is being referred to by the pronoun 'he' in the second sentence, they must refer back to the first sentence. Lexical cohesion on the other hand arises from the selection of vocabulary items and the semantic relationships between them. For example, *'I parked outside the* library*, and then went inside the building to return my books*', where cohesion is represented by the semantic relationship between the lexical items 'library', 'building' and 'books'. For automatic identification of these relationships it is far easier to work with lexical cohesion than reference because less underlying implicit information is needed to discover the relationship between the above pronoun and the word it references. Hence lexical cohesion is used as a linguistic device for investigating the discourse structure of texts and lexical chains have been found to be an adequate means of exposing this discourse structure. These lexical chains have many practical applications in IR and computational linguistics such as hypertext construction [8], automatic document summarization [9], the detection of malapropisms within text [10], as a term weighting technique capturing the lexical cohesion in a text [11], as a means of segmenting text into distinct blocks of self contained text [12]. For the purpose of this project we exploit three such applications:

1. We use lexical chains as a means of exploring and presenting the most prevalent topics discussed in news stories.
2. A valuable side effect of lexical chain creation is that the words of a text are automatically disambiguated.
3. Because lexical chains disambiguate words based on the context in which they occur, lexical chains also address two linguistic problems *synonymy* and *polysemy*, which hinder the effectiveness of traditional IR systems such as the vector space model.

3. CHAIN FORMATION ALGORITHM

In general the first task of an IR system is to execute a set of text operations (e.g. stemming, removal of stopwords) to reduce the complexity of a full text representation of a document into a more manageable set of index terms. Although these index terms are a subset of the original representation, their purpose is to adequately represent the semantic content of the original document in a more concise manner. This is a difficult NLP task, as natural language frequently does not obey the principle of compositionality where the meaning of the whole can be strictly determined from its parts. So in order to derive the correct representation of a text, we need to determine the interpretation of a word or phase in the context in which it occurs i.e. before the original text is manipulated into a set of index terms. The creation of lexical chains which is described below, aims to capture this additional textual information while still maintaining a manageable representation size.

Firstly each term contained in a particular document is dealt with in chronological order. Then each subsequent word is added to an existing lexical chain or becomes the seed of a new chain, in much the same manner as the clustering of documents. A stronger criterion than simple semantic similarity is imposed on the addition of a term to a chain, where terms must be added to the most recently updated (semantically related) chain. This favors the creation of lexical chains containing words that are in close proximity within the text, prompting the correct disambiguation of a word based on the context in which it was used. We use WordNet to determine the semantic relatedness between a candidate word and the words of a chain. If we view WordNet as a large semantic network of nodes (meanings) inter-related by semantic relations (meronymy, hyponymy, etc.), then finding a relationship

between two words in the chaining process involves activating the network of one node and observing the activity of the other in this activated network.

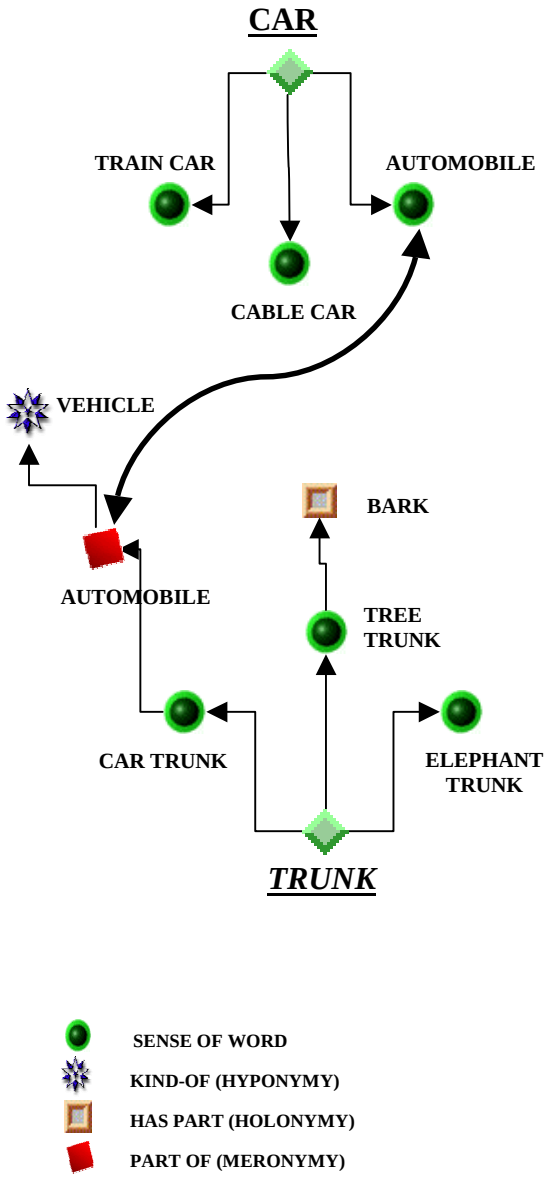

Figure 1: Shows expanded document terms 'car' and 'trunk' and their semantic relatedness.

So far we have talked abstractly about how to determine if a word is semantically related to a chain. To explain this fully it is first necessary to discuss the structure of the WordNet thesaurus, which is used to determine this semantic connection or closeness between words in a text. In WordNet, nouns, verbs, adjectives, and adverbs are arranged into synsets (group of synonymous words e.g. cat, feline, tabby), which are further organized into a set of lexical source files by syntactic category. In our case we are only interested in the noun index and data files, because the verb file in WordNet has no relation with the three other files (noun, adverb and adjective files), and the adverb file has only unidirectional relations with the adjective file. So each word in a particular document is searched for in the noun index file, if it is not found then we make the assumption that this word is not a noun and hence will play no further part in the chaining process. If the word is found then it will be represented by a unique set of synset numbers, where each synset number represents a particular sense associated with that word. Each synset number points to the position in the noun data file where words related to this sense of the word are stored with a gloss, and sample sentence using this word. Words related to a particular sense are associated with it by several different semantic relations, such as *hyponymy* (kind-of, lorry/vehicle), *hypernymy* (is-a, vehicle/car), *holonymy* (has-part, tree/branch) and *meronymy* (part-of, engine/car). As shown in Figure 1, each sense associated with a word is expanded using WordNet (in reality these senses and senses related to them are represented by synset numbers). This example of the chain formation process shows us that the word 'car' is related to the word 'trunk' by the fact that 'car trunk', one of the senses of 'trunk', is a meronymy of 'automobile' which is a possible sense of 'car'. In this way both words have been successfully disambiguated so all redundant senses belonging to each word are eliminated and 'car' is added to the chain containing 'trunk'. This chain may also contain other semantically related words pertaining to the topic of an automobile e.g. {car, trunk, engine, vehicle…}. The chain formation process is continued in this way until all the words in a particular document (in our case nouns) have been chained. Any words that remain unchained or ambiguous after this chaining process are eliminated from our chain word representation based on the following hypothesis:

'The occurrence of words in a text which fail to participate in the overall cohesive structure of a text (i.e. remain unchained) is purely coincidental. Consequently these words are considered irrelevant in describing the general topic of a document.'

This implies that our lexical chaining strategy also provides us with an automatic means of selecting the most salient features of a particular news story. So when all redundant words have been removed in this manner, all remaining chains are then merged into a single chain containing all the synset numbers from each individual chain involved in this process. This representation is a semantic representation as opposed to a syntactic representation (in the case of a 'bag of words' representation) because it contains concepts (i.e. synset numbers) rather than simple terms to represent the content of a document.

The final stage of our combined document representation strategy involves collecting all free text words for each document and storing them in a set of index files. So effectively our composite document representation used in the detection process (described in the next section) consists of two weighted vectors, a chain vector and an ordinary term vector, where both chain words and free text words are weighted simply in terms of the frequency in which they occur in a document.

4. DETECTION ALGORITHM USING THE FUSION METHOD

Online Detection or First Story Detection is in essence a classification problem where documents arriving in chronological order on the input stream are tagged with a 'YES' flag if they discuss a previously unseen news event, or a 'NO' flag when they discuss an old news topic. However unlike detection in a retrospective environment a story must be identified as novel before subsequent stories can be considered. The single-pass clustering algorithm bases its clustering methodology on the same assumption, the general structure of which is summarised as follows.

1. Convert the current document into a weighted chain word vector and a weighted free text vector.
2. The first document on the input stream will become the first cluster.
3. All subsequent incoming documents are compared with all previously created clusters up to the current point in time. A comparison strategy is used here to determine the extent of the similarity between a document and a cluster. In our IR model we use sub-vectors to describe our two distinct document representations. This involves calculating the closeness or similarity between the chain word vectors and free text vectors for each document/cluster comparison using the standard cosine similarity measure (used in this variation of the vector space model to compute the cosine of the angle between two weighted vectors). The data fusion element of this experiment involves the combination of two distinct representations of document content in a single cluster run i.e. *j* equals 2 in equation (1). So the overall similarity between a document *D* and a cluster *C* is a linear combination of the similarities for each sub-vector formally defined as:

$$Sim(D, C) = \sum_{j=1}^{k} w_j \cdot Sim(D_j, C_j) \qquad (1)$$

where *Sim(X, Y)* is the cosine similarity measure for two vectors *X* and *Y*, and *w* is a coefficient that biases the weight of evidence each document representation *j*, contributes to the similarity measure.

4. When the most similar cluster is found a thresholding strategy [13] is used to discover if this similarity measure is high enough to warrant the addition of that document to the cluster and the classification of the current document as an old event. If this document does not satisfy the similarity condition set out by the thresholding methodology then the document is declared as discussing a new event, and this document will form the seed of a new cluster.
5. This clustering process will continue until all documents in the input stream have been classified.

5. EXPERIMENTAL RESULTS

A number of experiments were conducted on the TDT-1 broadcast news collection [1]. The results of these experiments were used to observe the effects on first story detection when lexical chains are used in conjunction with free text as a combined document classifier. The main aim of the experiments was to determine if lexical chains are a suitable document representation when classifying news stories in the TDT domain. The official TDT evaluation requires that the system output is a declaration (a YES or NO flag) for each story processed. These declarations are then used to calculate two system errors percentage *misses* and *false alarms*. Misses occur when the system fails to detect the first story discussing a new event and false alarms occur when a document discussing a previously detected event is classified as a new event.

5.1 System Descriptions

Three distinct detection systems TRAD, CHAIN and LexDetect are examined in the following set of experiments. The TRAD system [13], our benchmark system in these experiments is a basic FSD system that classifies news stories based on the syntactic similarity between documents and clusters. The design of this system is based on a traditional vector space model which represents documents as a vector, each component of which corresponds to a particular word and who's value reflects the frequency of that word in the document. Classification of a new event occurs in a similar manner to that described in Section 4, the most important difference between the two methods is that a single free text representation is used to express document content, rather than a combined representation. A *Time Window* [13] of length 30 is employed in the TRAD, CHAIN and LexDetect systems.

The design of our second system LexDetect has been described in detail in sections 3 and 4. The dimensionality of LexDetect (80 words) remains static through out these experiments. Using the current method of lexical chain creation, just under 72% of documents contained greater than or equal to 30 chained words. We therefore normalized the length of chain word representations by imposing a chain dimensionality value

of 30 on all LexDetect schemes[2]. In theory it is possible to vary the length of the free text representation in our combined representation however in these experiments all schemes contain free text representations of length 50, since optimal performance is achieved for TRAD when dimensionality 50 is used. The final system parameter to be varied in these experiments is the weighting coefficient w_j used in equation (1). The design of our third system CHAIN like TRAD, involves the use of a singular document representation. However this document representation contains chain words only rather than free text terms, and so the dimensionality of the system must be 30.

5.2 The Data Fusion Experiment

From the results shown in Figure 2 (a Detection Error Tradeoff Graph where points closer to the origin indicate better overall performance), we deduce that a marginal increase in system effectiveness can be achieved when lexical chain representations are used in conjunction with free text representations in the detection process. In particular, we see that the miss rate of our FSD system LexDetect decreased with little or no impact to the false alarm rate of the system.

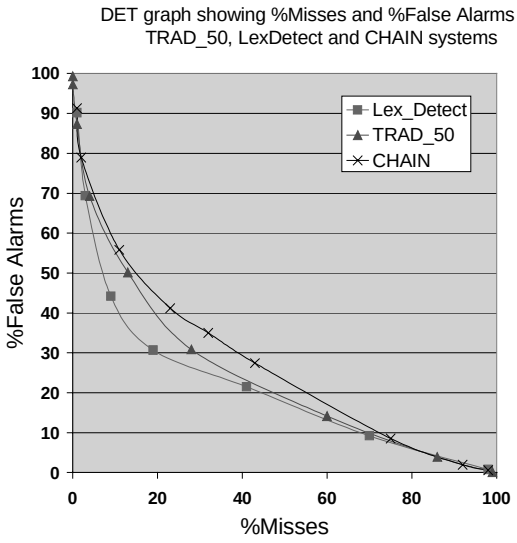

Figure 2: The effect on performance when a weighted combined document representation is used.

Optimal performance for the LexDetect system (as shown in Figure 2) was found when a weighted combination of evidence was used. This involved treating our free text representation as weaker evidence during the detection process. Results shown in Figure 3 contrast the effect on LexDetect performance when both the chain and free text representations are given equal weight (Lex) and when the weight of the free text representation is halved (LexDetect). This is an interesting result as similar experiments using composite document representations to improve search system performance based on ranking, only experienced optimal effectiveness when they allowed free text evidence to bias the retrieval process [14, 15]. This prompted us to question the necessity of the free text component of our composite representation, however results show that system performance degrades when this element of document content is excluded. This is due to the inability of WordNet to correlate the relationship between proper nouns and other semantically related concepts i.e. {Bill Clinton, US president}, which are often crucial in representing journalistic event identity because they reflect the '*who, what, where, when and how*' of a news story.

Our final experiment involves plotting TRAD_80 against LexDetect shown in Figure 4. The aim of this experiment is to prove that the increase in system effectiveness observed when a composite document representation is used can be attributed solely to the combination of evidence derived from our free text and chain representations rather than as a consequence of increasing the dimensionality of the system to 80 features. As the DET graph in Figure 4 shows, our LexDetect system still outperforms our TRAD system under conditions of equal dimensionality.

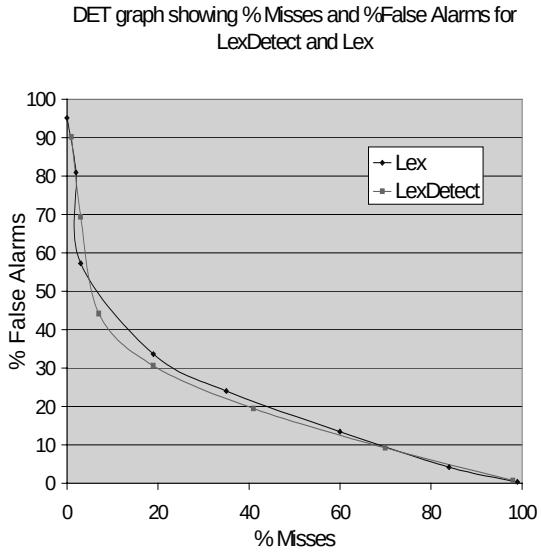

Figure 3: The effect on performance when equal weight is given to both representations (Lex) in contrast to a weighted combined document representation (LexDetect).

[2] An IR 'system' and an IR 'scheme' are used in this context to describe two different concepts. An IR system refers to the physical implementation of an IR algorithm, which can have various operational modes or various parameter settings. The same IR system may be used to execute different IR schemes by adjusting these parameters [20].

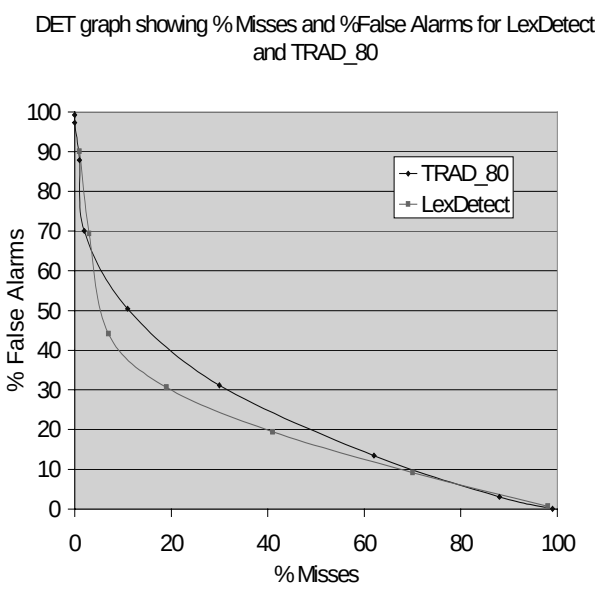

Figure 4: The effect on performance when equal dimensionality of 80 is given to both the LexDetect and TRAD systems.

6. CRITERIA FOR SUCCESSFUL DATA FUSION

In the previous section our results showed that when a chain word representation is used in conjunction with a free text representation of a document, improvements in FSD effectiveness are observed. However these results fail to provide any concrete reasoning as to why data fusion under these particular conditions work. There are many papers in the data fusion literature, which attempt to explain why certain data fusion experiments succeed where others have failed. Many of these papers look at the effects of combining specific sources of evidence such as the combination of rank retrieval lists, multiple searches or multiple queries. However Ng and Kantor [16] have tried to formulate some general preconditions for successful data fusion involving non-specific sources of evidence.

The first of these criteria is based on the *dissimilarity* between two sources of evidence.

1. *Dissimilarity*: Data fusion between operationally very similar IR systems may not give better performance.

To calculate the level of dissimilarity between our FSD systems described in Section 5, we now define two ratios based on the number of common relevant and common non-relevant tagged documents between two distinct systems. The number of relevant tagged documents, $|r_1 \cap r_2|$ is defined as the number of documents that were correctly classified (as a new or old event) by both systems. The total number of relevant documents, r_1+r_2 is the sum of the number of correctly classified documents for each system. $|n_1 \cap n_2|$ and n_1+n_2 are similarly defined in terms of the number of incorrectly classified documents returned by both systems (i.e. missed events or wrongly detected new events) as shown in equation 3.

$$R_{overlap} = \frac{|r_1 \cap r_2| \cdot 2}{r_1 + r_2} \qquad (2)$$

$$N_{overlap} = \frac{|n_1 \cap n_2| \cdot 2}{n_1 + n_2} \qquad (3)$$

The results for this experiment are shown in tables 1 and 2 below. We can see that in general the relevant document overlap $R_{overlap}$ between the pair-wise similarities of all four systems is between 85% and 92%, the most similar systems being not surprisingly our two TRAD schema which differ only in the length of their classifiers. The pair-wise similarities $N_{overlap}$ of all four systems regarding non-relevant document classifications exhibit a similar trend of high similarity between the TRAD and LexDetect systems. However the most important point to be taken from these sets of results regards the fact that our CHAIN and TRAD systems exhibit the lowest relevant and non-relevant document overlap of all our pair-wise comparisons. This is an important and encouraging result as it shows that our chain word representations (used in CHAIN) is sufficiently dissimilar to our simple 'bag of words' representation (used in TRAD) to contribute additional evidence to a combination experiment involving both these representations. In particular this satisfaction of Ng and Kantor's dissimilarity criteria explains why marginal improvements in system performance were observed in our data fusion experiment.

Table 1: Relevant document overlap between FSD systems.

$R_{OVERLAP}$	LexDetect	TRAD_50	TRAD_80	CHAIN
LexDetect	1			
TRAD_50	0.85	1		
TRAD_80	0.85	0.92	1	
CHAIN	0.56	0.52	0.53	1

Table 2: Non-relevant document overlap between FSD systems.

$N_{OVERLAP}$	LexDetect	TRAD_50	TRAD_80	CHAIN
LexDetect	1			
TRAD_50	0.67	1		
TRAD_80	0.68	0.82	1	
CHAIN	0.58	0.51	0.53	1

The second criteria defined for successful data fusion regards efficacy or the quality of the individual sources of evidence before they are combined in the data fusion process.

2. *Efficacy*: Data fusion between a capable IR system and a very incapable IR system may not give better performance.

In our data fusion experiment in Section 5 we observed that our CHAIN system was our worst performing FSD system. So as the efficacy criteria suggests a better performing chain word representation is needed before further improvements are observed in our combination system LexDetect.

7. FUTURE WORK

There are many factors which can affect the final chain word representation of a document, ranging from the greedy nature of the chaining algorithm, to the effects caused when varying degrees of freedom are used in this algorithm (i.e. system parameters such as the amount of activation used in WordNet). However the single biggest influence on the quality of the resultant lexical chains is the knowledge source used to create them. In other words the quality of our lexical chain formation is directly dependent on the comprehensiveness/complexity of the thesaurus used to create them. In the case of WordNet, there are a number of structural inadequacies that degrade the effectiveness of our chain representation:

1. Missing semantic links between related words.
2. Inconsistent semantic distances between different concepts.
3. Overloaded synsets such as 'being' which are connected to a large number of synsets. These types of synsets cause spurious chaining, where an unrelated word is added to a chain based on a weak yet semantically close relationship with one of these overloaded synsets (a special case of 2.).
4. No means of correlating the relationship between proper nouns and other noun phrases (see Section 5.2).
5. The level of sense granularity used to define word meanings in WordNet is often too fine for the chain formation process.

All of these factors play a part in reducing the effectiveness of the disambiguation process and the comprehensiveness and accuracy of the final chain representation. A number of these weaknesses are discussed in previous work on lexical chaining [8, 12]. However the last two cases are particularly important when considering the similarity between documents and clusters in the detection process. As explained in Section 6.2 lexical chains are an incomplete means of representing events in a topic detection application since they fail to contain information on the proper nouns involved in the discourse structure of the text.

The last case is more a comment on the unsuitability of WordNet as a knowledge source in this application rather than as a reference to any specific weakness in its design. For example consider two distinct documents which both contain the word 'city' in their respective chain representations. WordNet defines three distinct meanings or senses of this word:

⇒ An incorporated administrative district establish by a state charter.
⇒ A large densely populated municipality.
⇒ An urban center.

When disambiguating a word like 'city' in the chain formation process this level of sense distinction is unnecessary. In fact if our aforementioned documents have chosen two different yet closely related definitions of this word (i.e. different synset numbers) then these documents will be considered less related than they actually are. Other research efforts in the lexical chaining area have suggested 'cleaning' WordNet [8] of rare senses or using some additional knowledge source in the chaining process that could biases the suitability of certain senses in particular contexts[3]. In future work we hope to address this problem by considering the use of collocation information like noun pairs such as 'physician/hospital' or 'Gates/Microsoft' in the chain formation process. Using such information will help to smooth out the discrepancies in semantic distances between concepts and help detect missing semantic relationships between these concepts. This occurrence information could also reduce the sensitivity of the detection process to fine levels of sense granularity if such information was used when determining the similarity between two document representations. So effectively this technique would eliminate the need for a composite representation in the identification of novel events in a news stream. Instead the data fusion element of our system would involve supplementing our knowledge source WordNet with word co-occurrence information in the chain formation process.

8. CONCLUSIONS

A variety of techniques for data fusion have been proposed in IR literature. Results from data fusion research have suggested that significant improvements in system effectiveness can be obtained by combining multiple sources of evidence of relevancy such as document representations, query formulations and search strategies.

[3] Recent editions of WordNet now contain information on the probability of use of a word based on polysemy. WordNet researchers noted the direct relationship between the increase in the frequency of occurrence of a word and the number of distinct meanings it has. This frequency value could also be used in the 'cleaning' process.

In this paper we investigated the impact on FSD performance when a composite document representation is used in this TDT task. Our results showed that a marginal increase in system effectiveness could be achieved when lexical chain representations were used in conjunction with free text representations. In particular, we saw that the miss rate of our FSD system LexDetect, decreased with little or no impact to the false alarm rate of the system. When a weighted combination of evidence was used on the same system this improvement was even more apparent. From these results we deduced that using our chain word representation as stronger evidence in the classification process could lead to improved performance. Based on Ng and Kantor's dissimilarity criteria for successful data fusion we attributed the success of our composite document representation to the fact that a chain word classifier is sufficiently dissimilar to a simple 'bag of words' classifier to contribute additional evidence to a combination experiment involving both these representations. In future experiments, we expect an even greater improvement in FSD effectiveness as we continue to refine our lexical chain representation.

9. ACKNOWLEDGMENT

This project is funded by an Enterprise Ireland research grant [SC/1999/083].

10. REFERENCES

[1] R. Papka, J. Allan, Topic Detection and Tracking: Event Clustering as a basis for first story detection, Kluwer Academic Publishers, pp. 97-126, 2000.

[2] Y. Yang, T. Ault, T. Pierce, *Combining multiple learning strategies for effective cross validation*, the Proceedings of the 17th International Conference on Machine Learning (ICML), pp. 1167-1182, 2000.

[3] F. Walls, H. Jin, S.Sista, R. Schwartz, *Topic Detection in broadcast news*, In the proceedings of the DARPA Broadcast News Workshop, pp. 193-198, San Francisco, CA: Morgan Kaufman Publishers Inc, 1999.

[4] F. Fukumoto, Y. Suzuki, Event Tracing based on Domain Dependency, In the proceedings of the 23rd ACM SIGIR Conference, Athens, pp. 57-63, 2000.

[5] V. Hatzivassiloglou, L. Gravano, A. Maganti, *An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering*, In the proceedings of the 23rd ACM SIGIR Conference, Athens, pp. 224-231, 2000.

[6] J. Morris, G. Hirst, *Lexical Cohesion by Thesaural Relations as an Indicator of the Structure of Text*, Computational Linguistics 17(1), March 1991.

[7] M. Halliday, R. Hasan, *Cohesion in English*, Longman: 1976.

[8] S. J. Green, *Automatically Generating Hypertext By Comparing Semantic Similarity*, University of Toronto, Technical Report number 366, October 1997.

[9] R. Barzilay, M. Elhadad, *Using Lexical Chains for Text Summarization*, In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, 1997.

[10] D. St-Onge, *Detection and Correcting Malapropisms with Lexical Chains*, Dept. of Computer Science, University of Toronto, M.Sc Thesis, March 1995.

[11] M. A. Stairmand, W. J. Black, *Conceptual and Contextual Indexing using WordNet-derived Lexical Chains*, In the Proceedings of BCS IRSG Colloquium, pp. 47-65, 1997.

[12] M. Okumura, T. Honda, *Word sense disambiguation and text segmentation based on lexical cohesion*, In Proceedings of the Fifteen Conference on Computational Linguistics (COLING-94), volume 2, pp. 755-761, 1994.

[13] N. Stokes, P. Hatch, J. Carthy, *Topic Detection, a new application for lexical chaining?*, In the Proceedings of the 22nd BCS IRSG Colloquium on Information Retrieval, pp. 94-103, 2000.

[14] E. Fox, G. Nunn, W. Lee, *Coefficients for combining concept classes in a collection*, In the proceedings of the 11th ACM SIGIR Conference, pp. 291-308, 1988.

[15] J. Katzer, M. McGill, J. Tessier, W. Frakes, P. DasGupta, *A study of the overlap among document representations*, Information Technology: Research and Development, 1(4):261-274, 1982.

[16] K. Ng, P. Kantor, *An Investigation of the preconditions for effective data fusion in IR: A pilot study*, In the Proceedings of the 61th Annual Meeting of the American Society for Information Science 1998.