

# Judging relevance through identification of lexical chains

Steven Ngai and Matthew Holliman

{sngai,holliman}@stanford.edu

## ABSTRACT

Lexical chaining is a method for encapsulating the meaning of a document in so-called lexical chains. The presence and concentration of such chains can be used to judge similarities both within and between documents. We present a system that performs such chaining and, using a small corpus of news articles, compare its performance with that of a naive vector-space system. We then investigate the utility of such automated similarity judgments for information retrieval tasks such as generating inter-document links and processing queries.

## 1. INTRODUCTION

Today's Web users rely upon a combination of two techniques – searching (via queries) and “surfing” (via links) – to find the information they want. As the size of the Web continues to grow, it therefore becomes increasingly important to improve these two methods so that they return content that is interesting and relevant to the user.

One basic means of improving both of these operations is to try to provide more semantically-aware information retrieval (IR) systems. Semantically aware IR systems can improve queries by adding semantic understanding and sense disambiguation to query interfaces, generating a more precise picture of which documents are related to the query. Such systems can improve browsing by automatically inserting hyperlinks in pages to related content, allowing the user to browse to information of interest more easily. This latter approach is particularly advantageous over human insertion of links, given the time cost and inconsistencies associated with manual linking [4].

Ultimately, both approaches require the evaluation of document similarity as an estimate of document relevance. Many current automatic information retrieval systems [2] attempt to determine similarity through vector-space models. Such methods consider the text as a “bag of words.” Because of the simplicity of this model, these systems, unlike humans, are blind to complicating issues such as synonymy and polysemy, and further require that a word (or at least its stem) appear verbatim before being able to act upon it. Clearly it would help a computer to have, like humans, some insight into the meaning of the discourse at a higher level, as a “bag of ideas.” This insight is provided by the techniques of *lexical chaining* [6].

By taking into account some level of semantic meaning from texts when determining document similarity, we might expect results from a browsing/query system based on such principles to match the expectations of the system's human users more closely than those of simpler models. That is, the links inserted by such a system for browsing may better reflect the underlying semantic content of a document, and queries for documents might likewise incorporate some level of semantic understanding in the results they return.

To this end, in this study we have compared a standard baseline information retrieval methodology that the reader should be familiar with, the vector-space model, to an algorithm based on the lexical chaining work of Hirst and Green. Though our primary application for our relevance judgments, to which we will devote most of our attention, is the determination of relevant links to improve browsing, we will open up our discussion into the general performance of this lexical chaining algorithm, as well as other ways this system can be exploited. Such a system obviously plays an important part in IR, as it can form the backbone of a successful query, link-generation, or other information retrieval mechanism.

In the next section, we briefly review the basic principles behind lexical chaining. Section 3 discusses the implementation of the system. Section 4 evaluates our system's performance. In Section 5, we discuss some of the trade-offs we encountered in constructing this project. Finally, Section 6 concludes the paper.

## 2. LEXICAL CHAINING PRINCIPLES

Lexical chaining methods are founded on the idea that there are chains of discourse that run through a document, and further, that these chains contribute to the document’s meaning. For example, one might expect a hypothetical document about “virtual parenting” (absentee parenting through hi-tech devices) to contain chains relating to parents, work, children, and gadgets. If we imagine these chains as producing the words in the document, we can hope to “reverse engineer” the chains from the words with a proper knowledge of words and their meanings.

Any body of knowledge can be used for this purpose, e.g. a thesaurus would suffice. In this system, however, we have decided to follow in the footsteps of most recent researchers in the field including Hirst & St-Onge [7] and Green [4, 5], who use WordNet to supply this information. WordNet, a project of the Cognitive Science Laboratory at Princeton University, can be viewed as a graph of the English lexicon wherein the nodes represent the individual meanings of words and edges indicate semantic connections between words. The edges between word meanings (called synonym sets or synsets) are labeled to reflect the type of semantic connection.

After we have discovered the chains that run through a document, we have learned some information about the meaning of the document. There is no limit as to how one forms these chains (Al-Halimi [1] actually forms two-dimensional lexical *trees*) or what one can do with this information (for instance, Hirst [7] uses the results of lexical chaining to correct malapropisms). But as for our application, which seeks to judge similarities of meaning, we make the assumption that if two documents contain a similar concentration of chains, a human would judge the documents to be about the same topic and therefore similar. Likewise, if two paragraphs within a document contain a similar concentration of chains, we make the assumption that the paragraphs serve similar roles in the development of the narrative and therefore can be considered relevant.

A basic assumption is that similarity, which at least in computer science terms is a numerical judgment, roughly approximates relevance, which is a subjective human judgment based on information need.

## 3. SYSTEM OVERVIEW

The system is comprised of three components:

- a first **intra-document stage** that evaluates the similarities between units of text (in our system, these units are paragraphs);
- a second **inter-document stage** that uses the results of the first to evaluate inter-document similarity; and finally
- one of two possible **information retrieval stages** based on the computed similarity measures. The information retrieval approaches we examine include
  - a link generation stage based on the similarity metrics, to facilitate browsing of similar documents, and
  - a simple query front-end based on the same metrics, to facilitate search.

### 3.1. Intra-document similarity

The first part of the lexical chaining system operates on individual documents. The document is parsed to isolate words. If any word appears more than a threshold  $R$  times, the system sets up a chain for that word consisting of all of its synsets as all available parts of speech. Each synset encapsulates one sense of the word.

Clearly, if the  $R$  threshold requirement is to filter out noise effectively, it must scale with document length. Experimentally, we found that  $R = 2$  is best for documents of newspaper-article length. Section 5 explains why the proper choice of such initial conditions is important.

Now the many chains must undergo merging before the meaning of the document begins to emerge. For the purposes of chaining, all the different types of connections in WordNet boil down to two types:

- **Strong relations:** when two synsets are the same or related by hypernym/hyponym.
- **Regular relations:** when two synsets have between them a valid path of length less than  $N$ . For our system,  $N = 2$  (see Section 5 for a justification).

The process of chain merging involves two stages: first the system resolves all inter-chain strong relations, followed by regular relations. The process of resolving a relation of type  $X$  is as follows. Each existent chain is run against all other chains to determine how many  $X$  relations exist between its synsets and those of the other chain. The pair with the most such relations is merged into one chain; this signifies the combining of two related series of ideas into one overarching idea. All synsets that are not  $X$ -related to any synset in the other chain are eliminated; this causes any alternate meanings of words that are inconsistent with the growing pattern to be dropped.

Paragraph similarities are computed by comparing the concentration of chains in the paragraph. Each paragraph thus has associated with it a vector representing the density of all chains from the document in that paragraph; in our implementation, we have used the Dice function to compute inter-paragraph chain density correlations.\* If the similarity score between two paragraphs exceeds the mean inter-paragraph similarity by some threshold number of standard deviations, we declare that pair linkworthy.

In our system, the report that is returned for each article consists of an HTML document that allows the user to examine and “surf” the document by following either generated paragraph links or chains. In addition, the process also generates a “hash file,” to be explained in the next section.

### 3.2. Inter-document similarity

The second half of the system, which judges inter-document similarity, receives as its sole input the “hash file” document summaries generated in the document processing stage. Depending on the model, these hashes contain either synsets or words, followed by their frequencies in the document. Vector-space hashes (from our baseline comparison implementation) contain only words, since the model does not allow for anything else; chaining hashes may contain either synsets or words, reflecting whether the word in question had any WordNet synsets.

Regardless of the original model, two documents are created for each hash: a document proper and a linked document, consisting of all synsets immediately proximal (through a strong relation) to a synset in the document proper. (Because the notion of a proximal word is not defined, the linked document contains no words. It happens, then, that the linked document for the vector-space model is degenerate and always empty.) We treat the document proper and linked document as vectors, weighted by a formula that incorporates both normalization and inverse document frequency of words/synsets:

$$w = \frac{tf \times \log df}{\sum_i (tf_i \times \log df_i)^2} \quad (1)$$

where the summation occurs across all the synsets of the document under examination.

The similarity score for two articles  $A$  and  $B$  is found as

$$\text{sim}(A, B) = A_d \cdot B_d + A_d \cdot B_l + A_l \cdot B_d, \quad (2)$$

where  $X_d$  denotes document  $X$ 's document-proper vector and  $X_l$  denotes  $X$ 's linked vector.

Note that with this design, even if two lexically-chained documents contain words that cannot be found in WordNet, they will contribute to the similarity score in typical vector-space fashion. This provision is useful, since even though WordNet contains an abundance of technical terms and personalities, it cannot be expected to be complete and current. Our implementation of this stage thus addresses a shortcoming of Green's system, which ignores such words and thereby valuable indicators of similarity. Ours is a hybrid approach that hopes to take semantics into account, but failing that, falls back upon a vector space approach.

The output of this half of the system is an HTML document that contains a color-coded matrix describing the strength of similarities between documents. A separate script takes the matrix and a threshold value and generates a list of links that would be formed from each article.

## 4. EVALUATION

In our experiments, we have focused our attention particularly on inter-document similarity determination. Our basic reasoning for this is that we believe that inter-document similarity is generally of greater utility in “real-world” systems than intra-document similarity measures, since links between documents are typically more useful to users browsing documents than intra-document links. Moreover, such similarity measures are more pertinent to the application of running semantically-aware queries over large document sets.

---

\* $2(\sum_i x_i y_i) / (\sum_i x_i^2 + \sum_j y_j^2)$  for two normalized chain density vectors  $x$  and  $y$ .

aids-day.txt	celebration of World AIDS day; awareness of AIDS epidemics in third-world countries
cruises-ok.txt	Feds' insistence that cruises are still safe; summary of illness aboard two ships; origin and advice
disney-virus.txt	Magic's return home; the Amsterdam
disney-virus2.txt	continuing sickness; the Amsterdam
disney-virus3.txt	Magic's return home; the Amsterdam
hiv.txt	HIV cases to rise sharply in Britain; sub-Saharan epidemic; women
jeffords-bush.txt	Jeffords's criticism of Bush's treatment of the Clean Air Act and his general environmental record; secrecy in government
kenya-al-qaeda.txt	al Qaeda still as a possible suspect; summary of evidence and history
kenya-bombing.txt	Lack of a definite link to al Qaeda; ongoing evidence for responsibility and government response
kenya-forewarned.txt	US's forewarning by Australian government about possible attacks; aftermath and search for terrorists; criticism and technology
oil-spill.txt	impending environmental disaster caused by spill; recovery and news
oil-spill2.txt	impending environmental disaster caused by spill; recovery and news
protective-gear.txt	deficiencies in the equipment necessary to protect US soldiers against biochem agents
smallpox.txt	inoculation of first responders; opinions for and against
oil-aquarium.txt	oil from spill threatens saltwater aquarium; criticism
weapons-inspect.txt	Iraq's surprising, but reluctant, willingness to submit

Table 1: Corpus summary.

#### 4.1. Evaluation methodology

To evaluate the quality of our assessments of inter-document similarity and link insertion, we tested our system on sixteen news articles. These texts were all gathered from leading Internet news sources (AOL News [8], CNN [9], Washington Post [10], and the New York Times [11]) on November 30, 2002. Article titles, bylines, and section headings were stripped, but nothing else, including division into paragraphs, was changed. Table 1 lists brief descriptions of the articles, along with their corresponding file names for reference in the following text.

As true to news form, the articles are written in the “inverted pyramid” style, in which successive paragraphs fill in more and more of the story and background in increasing depth. Therefore, although different articles may begin with different information, they often converge into the same background information as the articles progress.

We then asked twenty-five Stanford students to find all similar pairs of articles from the sixteen we had selected. Of the 120 such pairs possible, participants were instructed to rank pairs coarsely as not related (0), weakly related (1), or strongly related (2). Participants were encouraged to use fractional scores. Each participant was asked to define similarity as he or she wished; we suggested, with the motivation of our system in mind, that such similarity consider whether a person interested in pursuing further information in one article would want to know about the other articles. The descriptive filenames were removed and the files shuffled to remove bias.

These sixteen articles were chosen to reflect a variety of different strengths of similarity. But whereas some articles, such as oil-spill and oil-spill2, are overwhelmingly similar, we expected that some people might also find more subtle links. For instance, the aids and disney-virus articles, and perhaps the smallpox article, deal with maladies; the smallpox, weapons-inspection, and al-qaeda articles all deal with terrorism; and the oil-spill and jeffords-bush articles deal with the environment. And not all obvious links are of the same strength: the three oil-spill articles all deal with the Spanish oil spill, but oil-aquarium spends over half the article focusing on the spill's effects on an aquarium.

The results are tabulated in Table 2, Table 3, and Table 4. Yet even in this relatively straightforward comparison task, they confirm what several researchers [3, 4] have found about human judgments of similarity: namely, they are rather inconsistent. The obvious matches—those pairs to which we gave related descriptive titles, such as kenya-\*.txt—received points from everyone. Some of the broader overarching topical connections that we expected, such as the environmental link between oil-\* and jeffords-bush, received some support. Several pairs made unanticipated appearances: some respondents, for instance, connected the kenya-\* stories to aids-day.

The number of non-zero comparisons returned by each subject, their granularity of judgment, and their treatment of transitivity could doubtless form the basis for an in-depth and very interesting study of human perception and behavior. For our purposes we will merely be content to consider *any* nonzero rating to indicate that a link should be placed between two articles. However, a real link-insertion system surely ought not cater to the linking choices of only one or two people by setting such a low threshold—in fact, a threshold could be set anywhere between 0.2 and 0.8 against the study results and still catch all the same principal matches. But for the purpose of giving our systems a bit of a challenge, however, we will examine how well our systems can match even the obscure similarities.

Run against the same set of sixteen articles, the two models came to excellent agreement with themselves and the human standard on the strongest links (0.2 similarity or higher, see charts). Either model appears to be quite suitable for finding strong links. Encouragingly, both models also agreed on a weak link between weapons-inspect and protective-gear, which was the strongest “non-obvious” similarity that the human participants found.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
aids (A)	1															
cruises (B)	0.08	1														
disney-virus (C)	0.08	0.9	1													
disney-virus2 (D)	0.08	0.9	1	1												
disney-virus3 (E)	0.08	0.92	0.98	0.98	1											
hiv (F)	0.84	0.08	0.08	0.08	0.08	1										
jeffords (G)							1									
kenya-al-qaeda (H)	0.08							1								
kenya-bombing (I)	0.04							0.98	1							
kenya-forewarned (J)	0.04							0.94	0.96	1						
oil-aquarium (K)			0.04	0.04	0.04		0.08				1					
oil-spill (L)		0.04	0.04	0.04	0.04		0.16				0.88	1				
oil-spill2 (M)				0.04	0.12		0.08				0.88		1			
protective (N)								0.04	0.04	0.04				1		
smallpox (O)	0.04	0.08	0.04	0.08	0.04	0.08	0.04	0.02	0.02	0.02				0.08	1	
weapons (P)								0.12	0.12	0.12				0.14		1

Table 2: Similarity determined by human participants.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
aids	1															
cruises	0.02	1														
disney-virus	0.03	0.38	1													
disney-virus2	0.01	0.41	0.53	1												
disney-virus3	0.02	0.41	0.80	0.53	1											
hiv	0.39	0.03	0.03	0.01	0.02	1										
jeffords-bush	0.02	0.02	0.02	0.01	0.01	0.02	1									
kenya-al-qaeda	0.02	0.02	0.01	0.01	0.02	0.01	0.02	1								
kenya-bombing	0.02	0.03	0.04	0.02	0.04	0.03	0.01	0.37	1							
kenya-forewarned	0.01	0.04	0.01	0.02	0.01	0.02	0.01	0.47	0.38	1						
oil-aquarium	0.02	0.03	0.02	0.01	0.03	0.02	0.03	0.03	0.02	0.02	1					
oil-spill	0.02	0.02	0.02	0.02	0.02	0.01	0.03	0.04	0.03	0.02	0.31	1				
oil-spill2	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.04	0.02	0.03	0.28	0.65	1			
protective	0.02	0.02	0.03	0.01	0.01	0.02	0.04	0.02	0.05	0.03	0.02	0.01	0.01	1		
smallpox	0.02	0.05	0.02	0.02	0.04	0.03	0.01	0.03	0.03	0.02	0.02	0.01	0.01	0.02	1	
weapons	0.02	0.03	0.03	0.02	0.03	0.03	0.04	0.05	0.06	0.04	0.03	0.02	0.02	0.18	0.02	1

Table 3: Similarity determined by vector-space model.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
aids	1															
cruises	0.00	1.02														
disney-virus	0.01	0.35	1.02													
disney-virus2	0.02	0.22	0.38	1.02												
disney-virus3	0.00	0.257	0.67	0.33	1.07											
hiv	0.15681	0.01	0.01	0.00	0.00	1.03										
jeffords-bush	0.01	0.01	0.01	0.01	0.01	0.01	1.12									
kenya-al-qaeda	0.01	0.00	0.05	0.02	0.00	0.01	0.00	1.01								
kenya-bombing	0	0	0.00	0.00	0.02	0.01	0.02	0.14	1.04							
kenya-forewarned	0.02	0.02	0.037	0.00	0.02	0.02	0.02	0.24	0.45	1.12						
oil-aquarium	0.00	0.02	0.02	0.00	0.02	0.04	0.03	0.02	0.00	0.02	1.07					
oil-spill	0	0.02	0.01	0	0.00	0.01	0.01	0.04	0.01	0.03	0.24	1.04				
oil-spill2	0.02	0.03	0.04	0.01	0.00	0.00	0.02	0.03	0.01	0.00	0.21	0.45	1.02			
protective-gear	0.00	0.01	0.02	0.01	0.01	0.00	0.02	0.03	0.05	0.02	0.02	0.01	0.01	1.02		
smallpox	0	0.01	0.03	0.04	0.01	0.11	0.01	0.03	0.03	0.03	0.01	0.03	0.02	0.00	1.08	
weapons-inspect	0	0.01	0.02	0.01	0.01	0.00	0.04	0.05	0.05	0.02	0.05	0.01	0.01	0.12	0.00	1.00

Table 4: Similarity determined by lexical chaining model.

Algorithm	Precision/Recall
Vector space model	46/70
Lexical chaining	47/70

**Table 5.** Comparison of precision and recall for vector space model and lexical chaining for the fixed point where precision equals recall.

## 4.2. Numerical evaluation

It turns out that the study participants assigned nonzero similarities to exactly 70 pairs of articles. Thus, to test the performance of our linking methodology, we let each system return documents to its 70th lowest similarity—create 70 links, as it were—and sought to measure precision/recall.<sup>†</sup> Under these criteria, the vector space model matched 46 links and missed 24; the lexical chaining model matched 47 different links and missed 23. These results give the chaining method a slight, albeit most likely statistically insignificant, advantage.

A different evaluation method, which allows us to use a finer granularity metric rather than an accumulation of discrete binary decisions, involves treating each similarity matrix as a vector and calculating the resultant cosine similarities. This measure allows us to compare the fit of each set of automatically judged similarities to the human-judged similarities. Letting  $S_H$ ,  $S_V$ , and  $S_L$  represent the three similarity judgment sets (human, vector space, and lexical chaining, respectively),

$$\begin{aligned}
\cos(S_H, S_V) &= [(S_H \cdot S_V) / (|S_H| |S_V|)] \\
&= 21.667 / (\sqrt{27.634} \sqrt{19.011}) \\
&= .9453 \\
\cos(S_H, S_L) &= [(S_H \cdot S_L) / (|S_H| |S_L|)] \\
&= 20.642 / (\sqrt{27.634} \sqrt{19.050}) \\
&= .8997.
\end{aligned}$$

By this latter metric the lexical chaining method performs slightly worse, although not much so. However, it seems likely that with an increased search depth—which due to its computational and memory requirements our present implementation cannot handle—and with further tuning of the chaining algorithm, the results could be improved significantly.

## 4.3. Qualitative effects

The primary motivation for using lexical chaining rather than the simpler vector-space model is its possibility of producing semantically richer results. Even despite the limited search depth in our implementation, we saw examples of lexical chaining incorporating both polysemy and semantic similarity in its result sets.

### 4.3.1. Polysemy

In order to demonstrate the ways that the two systems react to polysemy, we introduced three more documents into the system dealing with stars of some type: astro-stars, asian-stars, and stars-magazine. As can be inferred from the titles, “star” is used in its astronomical sense in the first article and in its popular media sense in the other two articles. The vector space model reasonably finds a weak (0.12) similarity between the latter two articles and, less reasonably, a below-threshold (values 0.05, 0.03) similarity between the first article and either of the second. These below-threshold similarities rise perceptibly above the surrounding noise (approximately 0.015) and, in fact, are the greatest below-threshold similarities that the latter articles have. This is because of the contribution of the word “star,” whose senses cannot be directly disambiguated with this system.

However, with the chaining system, we see that the corresponding similarities don’t rise above the noise at all (values 0.01, 0.00). In fact, if we examine the resulting synsets, we see that in joining with words like “Sun” in the astro-stars article, the word “star” retains only the synset Noun-7754660: “(astronomy) a celestial body of hot gases”; in asian-stars, it has become part of a chain including the words “actor,” “player,” and (Bruce) “Lee” and acquired the synset Noun-8024371: “a theatrical performer.” In the matching stage, these synsets do not line up, and no score is accorded.

<sup>†</sup>Note that under these conditions, we find the fixed point where precision precisely equals recall in order to compare the two schemes.

### 4.3.2. Synonymy and semantic similarity

A naive vector space system suffers also from a failure to identify words that are distinct but nonetheless semantically very similar. With lexically-chained documents, performing a linked comparison—which consists of looking for synsets in the documents that are a strong relation apart—allows us to capture those relations that are more subtle. A good example of this is seen when finding document similarity between `hiv.txt` and `smallpox.txt`. Although the articles' direct focus are unconnected, at a high level they are nonetheless conceptually related by the shared topic of medical maladies. This semantic similarity is not reflected by the vector space model, which finds a near-zero similarity between the two (0.02), but is successfully captured by the lexical chaining process, which finds a weak similarity of 0.09 between the articles.

## 5. DISCUSSION

Our system has required many tradeoffs involving time, space, and performance. The lexical chaining algorithm is inherently extremely computationally expensive, being of  $O(n^3)$  or greater complexity in the number of synsets or chains. Each synset or chain needs to be compared with every other ( $O(n^2)$ ), and this process is repeated anew after each merging until merging is complete ( $O(n)$  or higher). Thus the very use of the algorithm represents a tradeoff of both space and time in order to squeeze out what we have seen to be a tiny amount of improvement over simpler methods.

Furthermore, the combination of Java and JWNL gives rise to extremely poor performance. For instance, JWNL apparently reads from the dictionary files every time it is queried, leading to the system being completely I/O-bound. Green [4] reports that chaining a certain article in his C++-language implementation requires less than 0.3 seconds, but, depending on parameters, our system requires on the order of minutes to chain an average news article. Despite our best efforts to optimize the process and make tradeoffs primarily for time – even at one point trying to preserializing all of WordNet and keeping it entirely in memory – our testing was made more difficult by these extreme system bottlenecks.

Some of our tradeoffs are listed below:

- **Tries and wordlists for prefiltering** In response to WordNet's slow lookup times, we preprocessed WordNet's dictionary files with command-line tools to create wordlists for the four parts of speech that WordNet catalogs. These wordlists allowed us to consult a faster in-memory version before wasting time looking up words as the wrong part of speech. This resulted in a substantial time savings at the cost of memory. The use of a stopwords list consisting of common information-poor words also falls along these lines.
- **Optimizations to proximity search** Because the code will often sequentially require the proximities of several synsets to the same synset, we calculate and store the entire set of proximal synsets that are reachable from a given synset. When the code receives subsequent requests, it simply searches the set for the second synset. Furthermore, we store a finite history of ten such proximal sets for recently accessed synsets. These optimizations represent a tradeoff of time for space. (Obviously we could go further and store an infinite history, ensuring that we only processed a synset once, but this would be prohibitively expensive in terms of memory requirements.)
- **Choice of regular path length  $N$**  The choice of a moderate allowable path length is central to the success of the lexical chainer. It allows words with slightly different meanings to be unified into one chain. With no allowable paths, the chainer degenerates into a vector-space model, where the meanings of words cannot help to disambiguate the meaning of the text, and the text is only a series of word frequencies. As long as the moderate path length does not become so long as to begin allowing arbitrary meanings to be pulled together, increasing path length should set the chainer apart from the vector-space model, either for the bad or the good. However, as WordNet is a graph in which practically each node has neighbors, the choice of search depth has an approximately exponential influence on number of nodes visited. For instance, for the first noun sense of each of the random words shown in Table 6, we see an exponential effect with each successive figure growing by anywhere from  $2\times$  to  $4\times$  with each increase in search depth. (The first number in each table entry indicates the number of synsets reached, and the second number the time elapsed in seconds; about two seconds are due to the overhead of starting WordNet.) The time dependence is actually worse than the exponential effect shown here, since, as search path increases, each synset is judged to be increasingly mergeable with more synsets. Consequently, more chains will need to be merged, and as each is merged and evaluated again for further merging, the entire exponential process of comparison will need to be repeated. Therefore we have decided, given the slower performance of Java and of our WordNet, to allow a distance  $N$  of only two. This was a tradeoff of performance for time.
- **Starting with the right chains: Multi-Word Expressions (MWEs) and the Selection of  $R$**  The lexical chaining algorithm is rather sensitive to small changes in initial conditions. Since synsets are dropped when chains

	Search length			
	1	2	3	4
carrot	2/3	30/3	86/4	251/9
successor	1/2	8/3	353/10	1708/37
school	30/3	69/4	139/6	350/10
private_school	5/2	36/3	70/4	139/5
human	331/10	1644/35	4817/100	9234/180

**Table 6.** Synsets reached by search length. The first number indicates the number of synsets reached, and the second the time elapsed in seconds. About two seconds are due to the overhead of starting WordNet.

combine, the initial semantic proximity of the formant chains strongly affects the synsets left in resultant chains; those synsets in turn strongly affect those of the next stage; and so on.

One consequence is that we must guard against starting with irrelevant words. Since WordNet has thousands of MWEs, it is worth it, for instance, to search for and add synsets relating to “New\_York” than to add “New” and “York,” the latter of which is in the wrong country and which, for instance, might prevent a chain of American locations from forming. Our MWE check catches many useful MWEs. For instance, it recognized the following on-topic MWEs in the oil-spill article:

oil\_slick fuel\_oil break\_up member\_of\_parliament Prime\_Minister,

all but the first of which combined with other words to form meaningful chains. This represents a tradeoff of memory and time for performance.

The reader will also remember that we chose our word repetition threshold  $R = 2$ . For our application, we found that  $R = 3$  produces too few initial chains to effectively summarize the semantic contents of a document, and  $R = 1$  produces too much noise. When we ran our system with the latter condition, for instance, words that were unrelated to the main themes of the article and to each other began combining and forming chains with obscure semantic connections. This not only clutters up the chains that form a document but, as noted above, can significantly affect the outcome of chaining.

## 6. CONCLUSIONS

We have compared the performance of a baseline vector-space model with a lexical chaining-based algorithm for determining document similarity. The latter approach implements a hybrid scheme, where recognized words are treated with semantic awareness, while unrecognized terms are handled using a standard vector-space-like tf-idf model. In particular, we have compared the results of both similarity algorithms with human judgments to determine their applicability to automatic link generation, with potential for use as semantically aware query mechanisms.

Disappointingly, our study is somewhat inconclusive. On one hand, our results clearly indicate the potential of lexical chaining for incorporating some level of semantic awareness, e.g. polysemy and synonymy, into document similarity measures. On the other hand, the vector-space model is substantially simpler and yet still performs competitively in our experiments.

There are several questions that naturally arise for future investigation. Most obviously, one area for improvement is to tune the lexical chaining implementation. We believe that increasing the depth to which related relations are examined could provide useful gains in performance, and perhaps provide a clear improvement over the simple vector-space model. This would likely require a more efficient language, however.

Another avenue of possible exploration relating to this is to apply lexical chaining to support semantically-aware queries. For instance, we have tried prepending queries to documents before chaining, thereby determining to which paragraphs in the collection the queries are most closely related, as well as treating the query as a separate document from which chains and synsets are extracted before similarity determination. However, we have not sufficiently explored these simple ideas to determine for certain whether they can provide any gain over a naive vector-space-based query algorithm.

A final area of investigation, given the relative success of the simple vector-space model in our experiments, would be to compare the performance of an approach such as latent semantic indexing or a vector-space model that incorporates query expansion to that of lexical chaining. Such approaches may offer most of the benefits of lexical chaining at substantially lower complexity.

*Note:* Instructions for running the system are included in the submission’s README file.

## REFERENCES

1. R. Al-Halimi, R. Kazman, "Temporal Indexing through Lexical Chaining" in *WordNet: An Electronic Lexical Database*, C. Fellbaum (editor), Cambridge, MA: The MIT Press, 1998.
2. J. Allan, "Automatic hypertext construction." Ph.D. thesis, Cornell University, 1995.
3. D. Ellis, J. Furner-Hines, and P. Willett, "The creation of hypertext linkages in full-text documents: Parts I and II." Technical Report RDD/G/142, British Library Research and Development Department, Apr. 1994.
4. S. J. Green, "Automatically generating hypertext by computing semantic similarity." Ph.D. thesis, Department of Computer Science, University of Toronto, 1997.
5. S. J. Green, "Automated link generation: Can we do better than term repetition?" in *Proceedings of the 7th International World-Wide Web Conference*, Brisbane, Australia, Apr. 1998.
6. S. J. Green, "Lexical Semantics and Automatic Hypertext Construction." *ACM Computing Surveys* 31(4), Dec. 1999
7. G. Hirst, D. St-Onge, "Lexical Chains as representation of context for the detection and correction of malapropisms." in C. Fellbaum, editor, *WordNet: An electronic lexical database and some of its applications*, Cambridge, MA: The MIT Press, 1997
8. <http://www.aol.com/>
9. <http://www.cnn.com/>
10. <http://www.washingtonpost.com/>
11. <http://www.nytimes.com/>