Building hypertext links in newspaper articles using semantic similarity

Stephen J. Green
Department of Computer Science,
University of Toronto
Toronto, Ontario
CANADA M5S 3G4
sjgreen@cs.utoronto.ca

Abstract

We discuss an automatic method for the construction of hypertext links within and between newspaper articles. The method comprises three steps: determining the lexical chains in a text, building links between the paragraphs of articles, and building links between articles. Lexical chains capture the semantic relations between words that occur throughout a text. Each chain is a set of related words that captures a portion of the cohesive structure of a text. By considering the distribution of chains within an article, we can build links between the paragraphs. By computing the similarity of the chains contained in two different articles, we can decide whether or not to place a link between them.

1 Introduction

A recent survey, reported in Outing (1996), found that there were 1,115 commercial newspaper online services worldwide, 94% of which were on the World-Wide Web (WWW). Of these online newspapers, 73% are in North America. Outing predicts that the number of newspapers online will increase to more than 2000 in the next year.

The problem is that these services are not making full use of the hypertext capabilities of the WWW. The user may be able to navigate to a particular article in the current edition of an online paper by using hypertext links, but they must then read the entire article to find the information that interests them. These databases are "shallow" hypertexts; the documents that are being retrieved are dead ends in the hypertext, rather than offering starting points for explorations. In order to truly reflect the hypertext nature of the Web, links should to be placed within and between the documents.

As Westland (1991) has pointed out, manually creating and maintaining the sets of links needed for a large-scale hypertext is prohibitively expensive. This is especially true for newspapers, given the volume of articles produced every day. This could certainly account for the state of current WWW newspaper efforts. Aside from the time-and-money aspects of building such large hypertexts manually, there have been indications (see Ellis *et al.*, 1994a; Ellis *et al.*, 1996) that humans are inconsistent in assigning hypertext links between the paragraphs of technical documents. That is, different linkers disagree with each other as to where to insert hypertext links into a document.

The cost and inconsistency of manually constructed hypertexts does not necessarily mean that large-scale hypertexts can never be built. It is well known in the IR community that humans are inconsistent in assigning index terms to documents, but this has not hindered the construction of automatic indexing systems intended to be used for very large collections of documents. Similarly, we can turn to automatically constructed hypertexts to address the issues of cost and inconsistency.

In this paper, we will propose a novel method for building hypertext links within and between newspaper articles. We have selected newspaper articles for two main reasons. First, as we stated above, there is a growing number of services devoted to providing this information in a hypertext environment. Second, many newspaper articles have a standard structure that we can exploit in building hypertext links.

Most of the proposed methods for automatic hypertext construction rely on term repetition. The underlying philosophy of these systems is that texts that are related will tend to use the *same* terms. Our system is based on *lexical chaining* and the philosophy that texts that are related will tend to use *related* terms.

2 Background and previous work

2.1 Lexical chains

A *lexical chain* is a sequence of semantically related words in a text. For example, if a text contained the words *apple* and *fruit*, they would appear in a chain together, since *apple* is a type of *fruit*. Each word in a text may appear in only one chain, but a document will contain many chains, each of which captures a portion of the cohesive structure of the document. Cohesion is what, as Halliday and Hasan (1976) put it, helps a text "hang together as a whole". The lexical chains contained in a text will tend to delineate the parts of the text that are "about" the same thing. Morris and Hirst (1991) showed that the organization of the lexical chains in a document mirrors, in some sense, the discourse structure of that document.

The lexical chains in a text can be identified using any lexical resource that relates words by their meaning. Our current lexical chainer (based on the one described in St-Onge, 1995) uses the WordNet database (Beckwith *et al.*, 1991). The WordNet database is composed of synonym sets or *synsets*. Each synset contains one or more words that have the same meaning. A word may appear in many synsets, depending on the number of senses that it has. Synsets can be connected to each other by several different types of links that indicate different relations. For example, two synsets can be connected by a hypernym link, which indicates that the words in the source synset are instances of the words in the target synset.

For the purposes of chaining, each type of link between WordNet synsets is assigned a direction of up, down, or horizontal. Upward links correspond to generalization: for example, an upward link from *apple* to *fruit* indicates that *fruit* is more general than *apple*. Downward links correspond to specialization: for example, a link from *fruit* to *apple* would have a downward direction. Horizontal links are very specific specializations. For example, the antonymy relation in WordNet is given a direction of horizontal, since it specializes the sense of a word very accurately.

Given these types of links, three kinds of relations are built between words:

Extra strong An extra strong relation is said to exist between repetitions of the same word: i.e., term repetition.

Strong A strong relation is said to exist between words that are in the same WordNet synset (i.e., words that are synonymous). Strong relations are also said to exist between words that have synsets connected by a single horizontal link or words that have synsets connected by a single IS-A or INCLUDES relation.

Regular A regular relation is said to exist between two words when there is at least one *allowable* path between a synset containing the first word and a synset containing the second word in the WordNet database. A path is allowable if it is short (less than n links, where n is typically 4 or 5) and adheres to three rules:

1. No other direction may precede an upward link.
2. No more than one change of direction is allowed.
3. A horizontal link may be used to move from an upward to a downward direction.

When a word is processed during chaining, it is initially associated with all of the synsets of which it is a member. When the word is added to a chain, the chainer attempts to find connections between the synsets associated with the new word and the synsets associated with words that are already in the chain. Synsets that can be connected are retained and all others are discarded. The result of this processing is that, as the chains are built, the words in the chains are progressively sense-disambiguated. When an article has been chained, a description of the chains contained in the document is written to another file. For example, table 1 shows the chains that were recovered from an AP article (Associated Press, 1992) about the performance of the U.S. dollar.

As you can see from the table, the current WordNet-based implementation of the chainer is not perfect. For example, chain 16, which contains the words *yen*, *pound*, *franc*, and *pfennig*, should probably be combined with chain 1, which contains *dollar*. These sorts of errors are to be expected when one uses a technique which is meant to work quickly (chaining this article takes 0.3 seconds). The expectation is that the uses to which we will put lexical chains will be able to cope with the "noise" in the data.

Table 1: Lexical chains from an AP story about the U.S. dollar's performance.

C	Word	Syn	C	Word	Syn	C	Word	Syn
1	dollar (7)	73240		observer (1)	59132	14	given (1)	45683
		73261		participant (1)	62489	15	high (1)	36227
	currency (2)	73203		brother (1)	60068			56231
	monetary_system (1)	73143			60071		stop (1)	34807
	sterling (1)	73200		trader (2)	63594			56493
2	fell (2)	20425		investor (1)	61585	16	yen (1)	74507
	withdrawal (1)	20453		french (1)	59471		pound (1)	74355
	cutting (1)	21099		swiss (1)	59473			74411
		21476		the_british (1)	59468			74414
	move (1)	20650		norman (1)	59367			74418
		20874		chairman (1)	62590			74420
	trading (4)	19862		john (1)	63848			74422
	trade (1)	23743	6	rose (2)	42329			74424
	market (1)	23761			54005			74426
	action (1)	19713			69460		franc (1)	74174
	economy (1)	20313	7	yesterday (1)	79599			74234
	credit (1)	19712			80100		pfennig (1)	74366
		19807	8	speculation (1)	48675	17	read (1)	47864
	rally (1)	19923		news (1)	48105	18	vice_president (1)	63720
	close (1)	47235			48251		chancellor_of_the_exchequer (1)	60211
	direction (1)	50358		remark (2)	48607	19	weekend (1)	79663
	interest (2)	21244		comment (2)	48607		week (1)	79661
	ease (1)	41201		offer (1)	50341		less (1)	73829
	profit (1)	43132	9	let (1)	21849		correction (1)	74131
	magnitude (1)	42859	10	rate (3)	72934		friday (2)	79631
3	major (1)	61881		interest_rate (1)	72911	20	bracket (1)	48941
4	tokyo (1)	57094		consideration (1)	72925			48944
	london (2)	57008	11	policy (1)	45713		mark (2)	48825
5	canadian (1)	59296			47644	21	federal_reserve_board (1)	55404
	new_york (4)	57548			48154		committee (1)	55407
	italian (1)	59380	12	light (1)	46597		central_bank (1)	55467
	european (1)	59256		position (1)	46594	22	side (1)	57979
	german (2)	59545	13	lower (1)	32993		decline (1)	57980

2.2 Automatic hypertext construction

2.2.1 A link apprentice

Bernstein (1990) proposes what he calls a *link apprentice*. This is a software tool that can be used to examine the draft version of a hypertext and propose links that a human editor or author can either accept or reject. The apprentice that he proposes is a “shallow” one, considering only lexical equivalence. While an author is working on a particular node, the system scans the rest of the nodes in the hypertext for nodes that are similar to the current one. The apprentice is intended for “compact, independent hypertext documents” (Bernstein, 1990, p. 213) such as textbooks or training manuals, and would probably not fare so well in a wider domain where there is a large amount of text to be linked and little opportunity for human involvement.

2.2.2 Unrestricted hypertext construction

More recently, Allan (1995) has been working on the automatic construction of hypertexts using the vector space model of the SMART information retrieval system (Salton, 1989). His work is significant in that it is intended to work on unrestricted collections of documents, rather than on single documents.

Document similarity is determined by considering the similarity of the two vectors that represent the documents. This global (i.e., document level) restriction can be extended to a local restriction in order to defeat the problem of polysemy. If two documents show a sufficient similarity, they can then be broken down into pieces (usually sentences). Each piece of one document can then be compared to each piece of the other. If there is a common usage of words between these pieces of the document, then they are assumed to be using the same words in the same senses. A link can then be placed between the two documents.

3 Building links within an article

3.1 Lexical chains as an indicator of structure

As part of their work, Morris and Hirst (1991) demonstrated that the structure of the lexical chains in a document corresponds to the structure of the document itself. In other words, the lexical chains will tend to delineate the parts of a document that are “about” the same topic. Due to the difficulty of building the lexical chains by hand, they were unable to test whether this is the case for a large number of texts. If the lexical chains *do* indicate the structure of the document, then they are a natural tool to use when attempting to build a hypertext representation of a document. If we are using documents that have a strict structure, such as newspaper articles, then the chains should prove sufficient to build *intra-article* links, that is, hypertext links within an article.

In the original work, Morris and Hirst attempted to define a straightforward, one-to-one mapping between the lexical chains in a document and the structural units of the document. It seems that this approach may have been a bit too straightforward, and so we will be presenting a more detailed approach. Our approach is similar to Morris and Hirst’s in that we assume that the parts of a document that have the same lexical chains are about the same thing, but we are willing to consider that a particular unit of a document’s structure may be indicated by the presence of more than one chain.

3.2 Analyzing the lexical chains

Newspaper articles are written so that one may stop reading at the end of any paragraph and feel as though one has read a complete unit. For this reason, it is natural to choose to use paragraphs as the nodes in our hypertext. Table 1 showed the lexical chains recovered from a news article about the performance of the U.S. dollar. Figure 1 shows the first and fifth paragraphs of this article with the words that participate in lexical chains tagged with their chain numbers. We will use this particular article to illustrate the process of building intra-article links.

The first step in the process is to determine how important each chain is to each paragraph in an article. We judge the importance of a chain by calculating the fraction of the content words of the paragraph that are in that chain. We refer to this fraction as the *density* of that chain in that paragraph. The density of chain c in paragraph p, $d_{c,p}$, is defined as:

$$d_{c,p} = \frac{w_{c,p}}{w_p}$$

The U.S. dollar[1] fell[2] against other major[3] currencies[1] in thin trading[2] yesterday[7] amid renewed speculation[8] that interest rates[10] still might ease[2].

The dollar[1] sank as low as 1.4580 German[5] marks[19], with observers[5] attributing most of the decline[22] to Mr. Greenspan's remarks[8], although many noted that the currency[1] was ripe for a downward correction[19] after an unchecked 10-pfennig[16] rally[2] last week[19].

Figure 1: Two paragraphs of an article tagged with chain numbers.

where $w_{c,p}$ is the number of words from chain c that appear in paragraph p and w_p is the number of content words (i.e., words that are not stop words. in p. For example, if we consider paragraph 1 of our sample article, we see that there are two words from chain 1. We also note that there are 13 content words in the paragraph. So, in this case, the density of chain 1 in paragraph 1, $d_{1,1}$ is:

$$d_{1,1} = \frac{2}{13} = 0.154$$

The result of these calculations is that each paragraph in the article has associated with it a vector of chain densities, with an element for each of the chains in the article. These *chain density vectors* for our sample article are shown in table 2. Note that an empty element indicates a density of 0.

Table 2: The chain density vectors for the U.S. dollar article.

	Paragraph										
Chain	1	2	3	4	5	6	7	8	9	10	11
1	0.154		0.125	0.100	0.105	0.050	0.105		0.091		0.111
2	0.308	0.161		0.300	0.053	0.100	0.211	0.062	0.091	0.125	
3	0.077										
4							0.105	0.062			
5		0.097	0.125		0.105	0.100	0.053	0.125	0.136	0.125	0.333
6							0.053	0.125			
7	0.077										
8	0.077	0.065			0.053	0.050			0.091		
9		0.032									
10	0.077	0.065	0.125								0.056
11									0.045		
12						0.050			0.045		
13			0.125								
14				0.100							
15		0.032				0.050					
16					0.053		0.158	0.125			0.111
17				0.100							
18						0.050			0.045		
19		0.032	0.125		0.105		0.053	0.125			
20					0.053						0.111
21		0.065							0.045		
22					0.053					0.125	
Chain Words	10	17	5	6	11	9	14	10	13	3	13
Content Words	13	31	8	10	19	20	19	16	22	8	18

3.3 Determining paragraph links

As we said earlier, the parts of a document that are about the same thing, and therefore related, will tend to contain the same lexical chains. Given the chain density vectors that we computed above, we need to develop a method to determine the similarity of the sets of chains contained in each paragraph.

3.3.1 Calculating paragraph similarity

Once we have the set of (possibly weighted and normalized) chain density vectors, the second stage of paragraph linking is to compute the similarity between the paragraphs of the article by computing the similarity between the chain density vectors representing them. We can compute the similarity between two chain density vectors using any one of 16 similarity coefficients that we have taken from Ellis et al. (1994b). These 16 similarity coefficients include both distance coefficients (where smaller numbers indicate a greater similarity) and association coefficients (where greater numbers indicate a greater similarity).

This similarity is computed for each pair of chain density vectors, giving us a symmetric $p \times p$ matrix of similarities, where p is the number of paragraphs in the article. From this matrix we can calculate the mean and the standard deviation of the paragraph similarities.

Table 3 shows the 11×11 symmetric similarity matrix for our example article. This particular similarity matrix was calculated using the Dice association coefficient with no weighting and no normalization. Since we used an association coefficient, larger numbers indicate a greater similarity.

Table 3: An 11×11 similarity matrix for the U.S. dollar article, calculated using the Dice coefficient of similarity.

Par	1	2	3	4	5	6	7	8	9	10	11
1		0.6171	0.2621	0.8220	0.3854	0.4849	0.6699	0.1811	0.5055	0.4072	0.1457
2			0.3748	0.5660	0.5193	0.7341	0.5407	0.4321	0.7143	0.6592	0.3545
3				0.1262	0.6305	0.3390	0.2959	0.4211	0.4378	0.2500	0.5450
4					0.3150	0.4590	0.6707	0.1970	0.4237	0.4494	0.0819
5						0.5951	0.5660	0.6164	0.6784	0.5601	0.5897
6							0.4777	0.3647	0.8642	0.6299	0.4233
7								0.6964	0.4741	0.4488	0.3728
8									0.3727	0.4000	0.5015
9										0.5767	0.5476
10											0.4206

Number of pairs:	55
Average similarity:	0.4763
Std. Deviation:	0.1709

3.3.2 Deciding on the links

The next step is to decide which paragraphs should be linked, on the basis of the similarities computed in the previous step. We make this decision by looking at how the similarity of two paragraphs compares to the mean paragraph similarity across the entire article. Each similarity between two paragraphs i and j, $s_{i,j}$, is converted to a z-score, $z_{i,j}$. If two paragraphs are more similar than a threshold given in terms of a number of standard deviations, then a link is placed between them. The result is a symmetric adjacency matrix where a 1 indicates that a link should be placed between two paragraphs.

Continuing with our example, consider $s_{1,5} = 0.3854$. We know that the mean paragraph similarity is 0.4763 and that the standard deviation in paragraph similarity is 0.1709. We can then determine that:

$$z_{1,5} = \frac{0.384 - 0.4763}{0.1709} = -0.54$$

That is, the similarity of paragraphs 1 and 5 is 0.54 standard deviations closer to 0 than the mean. For a similarity threshold of 1.0, we would *not* link these paragraphs, as they are not sufficiently similar. Table 4 shows the adjacency matrix that is produced when a threshold of 1.0 is used to compute the links from the similarity matrix in figure 3.

Table 4: Adjacency matrix for the U.S. dollar article.

Par	1	2	3	4	5	6	7	8	9	10	11
1		0	0	1	0	0	1	0	0	0	0
2			0	0	0	1	0	0	1	1	0
3				0	0	0	0	0	0	0	0
4					0	0	1	0	0	0	0
5						0	0	0	1	0	0
6							0	0	1	0	0
7								1	0	0	0
8									0	0	0
9										0	0
10											0

3.3.3 Generating HTML

Once we have decided which paragraphs should be linked, we need to be able to produce a representation of the hypertext that can be used for browsing. In the current system, there are two ways to output the HTML representation of an article. The first simply displays all of the links that were computed during the last stage of the process described above. The second is more complicated, showing only some of the links. The idea is that links between physically adjacent paragraphs should be omitted so that they do not clutter the hypertext.

4 Building links between articles

While it is useful to be able to build links *within* articles, for a large scale hypertext, links also need to be placed *between* articles. Recall from section 2.1 that the output of the lexical chainer is a list of chains, each chain consisting of one or more words. Each word in a chain has associated with it one or more synsets. These synsets indicate the sense of the word as it is being used in this chain. Table 5, shows some of the chains recovered from a 1992 article about the debate preceding the referendum on the Charlottetown Accord on the Canadian constitution. The numbers in parentheses show the number of occurrences of a particular word. Table 6 shows a portion of the chains from another article about the referendum.

Our aim is to build hypertext links between articles that will account for the fact that two articles that are about the same thing will tend to use similar (although not necessarily the same) words. For example, the set of chains in table 5 contains the word *proponent*, while the set in table 6 contains the synonym *advocate*. Similarly the first set of chains includes the words *negotiation* and *bargaining*, while the second contains *talks*. More distant relations between words should also be taken into account when building links. For example, the first set of chains contains the word *vote* while the second set contains the related word *referendum* (a kind of vote).

We can build these *inter-article* links by determining the similarity of the two *sets* of chains contained in two articles. In essence, we wish to perform a kind of cross-document chaining.

4.1 Synset weight vectors

We can represent each document in a database by two vectors. Each vector will have an element for each synset in WordNet. An element in the first vector will contain a weight based on the number of occurrences of that particular synset in the words of the chains contained in the document. An element in the second vector will contain a weight based on the number of occurrences of that particular synset when it is one link away from a synset associated with a word in the chains. We will call these vectors the *member* and *linked synset vectors*, or simply the member and linked vectors, respectively.

The weight of a particular synset in a particular document is not based solely on the frequency of that synset in the document, but also on how frequently that term appears throughout the database. The synsets that are the most heavily

Table 5: Lexical chains from an article about the 1992 referendum.

C	Word	Syn	C	Word	Syn	C	Word	Syn
3	minister (2)	60547		canadian (1)	59296		audience (1)	50268
		62036		charlottetown (2)	56923		speaking (1)	50133
	someone (1)	19677		canada (1)	56897		saying (1)	50294
	proponent (1)	59645		town (1)	56532	8	media (1)	46755
	loser (1)	61835		wall (1)	58333		radio (2)	46819
		61836		brick (1)	33318	16	government (1)	23871
	winner (1)	59612		table (2)	52008		vote (1)	20271
	nation (1)	54849		provisions (1)	52043		referendum (1)	20269
	people (4)	54284		special (2)	52086		going (1)	19729
	french (1)	59471	7	mistake (1)	48624			19749
	opponent (1)	59642		message (1)	47901		course (1)	19714
		62258		fact (1)	48086		express (1)	23799
	negotiator (1)	62150		appeal (1)	47593	19	years (1)	79589
	reporter (1)	62807		talk_show (1)	48004			80023
	premier (4)	60209		bit (1)	49143		week (1)	79661
		60210		**negotiation** (2)	50282		yesterday (2)	79599
	tough (2)	61447		**bargaining** (1)	50286		month (1)	79829
	quebec (7)	56924		interview (3)	50268		thursday (1)	79630

Table 6: Lexical chains from a related article.

C	Word	Syn	C	Word	Syn	C	Word	Syn
1	old (1)	79446		document (1)	73284		equal (1)	59131
	time (1)	19693		draft (2)	73160		james (1)	64012
	no. (1)	73875		sign (1)	48708			64013
	proof (1)	74863			48716		native (1)	59117
	yesterday (3)	79599	3	join (1)	54448		heavyweight (1)	60348
	day (1)	79595		national (1)	59127			61197
	today (2)	79598		people (2)	54284		accused (1)	59609
	sept (1)	79865		group (1)	19698	4	answer (1)	50484
	week (1)	79505		population (1)	54876		question (1)	50461
		79506		country (1)	54849		resignation (1)	50714
	authority (1)	42621		blind (1)	54288		accord (6)	50388
	power (3)	43265		man (1)	54283		dispute (1)	50408
	damn (1)	43047		business (1)	55647		arguing (1)	50412
2	strategy (3)	45730		justice (1)	61628		drafting (1)	22044
	tactic (2)	45729		official (1)	62223			23111
	fact (1)	48086		politician (3)	62524		office (1)	21928
	agenda (1)	45752			62525		helping (2)	24175
	allegation (1)	50638		leader (1)	59122		referendum (5)	20269
	charge (1)	50630		prime_minister (3)	60210		vote (1)	20271
	appeal (1)	47593		premier (3)	60210		election (1)	20270
	praise (1)	48306		**advocate** (1)	59645		fight (1)	24052
	interview (1)	50268		separatist (1)	63052		fighting (1)	24052
	debate (4)	50258		aboriginal (1)	59204		protest (1)	24073
	talks (3)	50282		minister (2)	62036		express (1)	23799
	speech (3)	50130		doer (2)	59620		hard_sell (1)	23773
	rhetoric (1)	49971		strategist (2)	63382		sideline (1)	21245
		49982						

weighted in a document are the ones that appear frequently in that document but infrequently in the entire database. The weights are calculated using the function that is used to weight term vectors in SMART (Salton and Allan, 1993):

$$w_{ik} = \frac{sf_{ik} \cdot \log(N/n_k)}{\sqrt{\sum_{j=1}^{s}(sf_{ij})^2 \cdot (\log(N/n_j))^2}}$$

where sf_{ik} is the frequency of synset k in document i, N is the size of the document collection, n_k is the number of documents in the collection that contain synset k, and s is the number of synsets in all documents. Note that this equation incorporates the normalization of the synset weight vectors.

The weights are calculated independently for the member and linked vectors. We do this because the linked vectors introduce a large number of synsets that do not necessarily appear in the original chains of an article, and should therefore not influence the frequency counts of the member synsets. Thus, we make a distinction between strong links that occur due to synonymy, and strong links that occur due to IS-A or INCLUDES relations. The similarity between documents is then determined by calculating three similarities (shown by the lines in figure 2):

1. The similarity of the member vectors of C_1 and C_2;
2. The similarity of the member vector of C_1 and linked vector of C_2; and
3. The similarity of the linked vector of C_1 and the member vector of C_2.

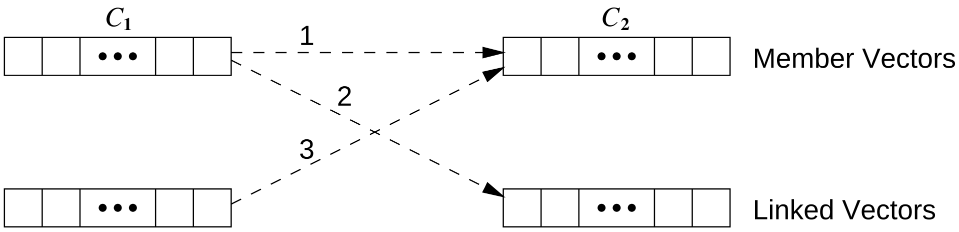

Figure 2: Computing chain similarity.

Clearly, the first similarity measure (the *member-member* similarity) is the most important, as it will capture extra-strong relations as well as strong relations between synonymous words. The last two measures (the *member-linked* similarities) are less important as they capture strong relations that occur between synsets that are one link away from each other.

If we enforce a threshold on these measures of relatedness, we can capture our requirement for multiple connections, since each element of the vectors will contribute only a small part of the overall similarity. We can calculate this similarity for all pairs of chains from two articles, and if there are a certain number of pairs that are more similar than our threshold, we can then say that the two articles should be linked.

4.2 Building inter-article links

Once we have built a set of synset weight vectors for a collection of documents, the process of building links between articles is relatively simple. Given an article that we wish to build links from, we can compute the similarity between the article's synset weight vectors and the vectors of all other documents. Documents whose member vectors exceed a given threshold of similarity will have a link placed between them. Our preliminary work shows that a threshold of 0.2 will include most related documents while excluding many unrelated documents.

This is almost exactly the methodology used in vector-space IR systems such as SMART, with the difference being that for each pair of documents we are calculating three separate similarity measures. The best way to cope with these multiple measurements seems to be to rank related documents by the sum of the three similarities. The sum of the three similarities can lie, theoretically, anywhere between 0 and 3. In practice, the sum is usually less than 1. For example, the average sum of the three similarities when running the vectors of a single article against 5,592 other articles is 0.039.

4.3 Testing inter-article links

Although there is a need for a full scale evaluation to determine the usefulness of our machine-generated links for IA tasks, by testing our linker against a set of reference queries we can make an initial determination of the capabilities of our linking methodology.

Our test involves taking a set of articles that are known to be related and seeing what connections are made between them. Such a set can be taken from the data used for the Text Retrieval Conference (TREC) (Harman, 1994). The object of TREC is a head-to-head evaluation of IR systems. Participating sites are provided with approximately 2GB of data comprising AP wire stories, six years of the *Wall Street Journal*, *San Jose Mercury News* articles, Ziff-Davis magazine articles, U.S. Department of Energy abstracts, U.S. Federal Register articles, and U.S. Patent abstracts.

TREC participants are given a set of *topics* which specify an information requirement. These topics are used in training the IR systems. Participants are also provided relevance judgments, detailing which documents are relevant to which topics. From this set of judgments we can select a set of articles to use in a preliminary evaluation of our inter-article linking methodology. We wish to determine whether articles that are relevant to the same topic will be linked, while articles that are relevant to different topics will not be linked.

4.3.1 Selecting topics

For our evaluation, we selected six topics from the 50 available. Table 7 shows the descriptions of each topic. Notice that the topics fall into two distinct groups, those about satellite systems (113–4), and those about cancer treatments (121–4). If our linking methodology works perfectly, then we would expect that documents that are relevant to one topic would never be linked to documents relevant to a different topic. Unfortunately, this may be too much to expect, especially given that some documents are relevant to more than one topic. A more realistic expectation would be that documents relevant to the "satellite" topics are not linked to the "cancer" topics, and *vice-versa*.

Table 7: Descriptions of topics used for evaluation.

Topic	Description
113	Document will report on non-traditional applications of space satellite technology.
114	Document will provide data on launches worldwide of non-commercial space satellites.
121	Document will discuss the life and death of a prominent U.S. person from a specific form of cancer.
122	Document will report on the research, development, testing, and evaluation (RDT&E) of a new anti-cancer drug developed anywhere in the world.
123	Document will report on studies into linkages between environmental factors or chemicals which might cause cancer, and/or it will report on governmental actions to identify, control, or limit exposure to those factors or chemicals which have been shown to be carcinogenic.
124	Document will report on innovative approaches to preventing or curing cancer.

For our evaluation, we excluded documents from the Department of Energy, Patent Office, and Federal Register corpora, since they do not follow traditional newspaper style. We were left with 2406 documents relevant to one or more of these topics. Rather than computing the similarity of all document pairs, a computationally expensive task, we decided to use a clustering technique to find groups of documents that could be linked to one another. The clustering technique used is the same as that used in the SMART system. This technique requires only $O(n)$ time, as opposed to the $O(n^2)$ time for computing all document similarities.

4.3.2 Clustering runs

We performed three separate runs of the clustering algorithm, using similarity thresholds of 0.1, 0.15, and 0.2. The results are shown in tables 8 through 10. We distinguish four kinds of clusters in the results:

Unit A "cluster" containing a single document vector.

With Same Topic A cluster containing more than one document vector where the vectors are from articles relevant to a single topic.

With Similar Topics A cluster containing more than one document vector where the vectors are from articles relevant to topics in the same group.

With Different Topics A cluster containing vectors from articles relevant to topics in different groups.

The percentage measures indicate the percentage of the documents relevant to each topic that are included in the number of clusters.

Table 8: Clustering TREC articles with a threshold of 0.1.

	Unit		With Same Topic		With Similar Topics		With Other Topics	
Topic	Clusters	Percent	Clusters	Percent	Clusters	Percent	Clusters	Percent
113	29	5.9%	63	63.3%	115	8.2%	80	22.7%
114	12	3.9%	9	29.9%	115	44.4%	67	21.9%
121	70	13.3%	94	56.7%	56	19.0%	41	11.0%
122	10	6.1%	5	7.4%	32	58.9%	16	27.6%
123	48	8.1%	53	48.8%	62	34.2%	25	8.9%
124	30	9.4%	8	6.9%	75	69.9%	25	13.8%

Number of clusters: 626

Table 9: Clustering TREC articles with a threshold of 0.15.

	Unit		With Same Topic		With Similar Topics		With Other Topics	
Topic	Clusters	Percent	Clusters	Percent	Clusters	Percent	Clusters	Percent
113	111	22.7%	81	56.7%	111	8.4%	56	12.2%
114	30	9.6%	29	53.1%	111	23.8%	52	13.5%
121	206	39.1%	81	45.2%	45	11.8%	16	4.0%
122	30	18.4%	14	25.8%	28	44.8%	6	11.0%
123	104	17.4%	71	46.0%	52	28.7%	22	7.9%
124	71	22.3%	13	17.9%	69	53.3%	19	6.6%

Number of clusters: 1008

Table 10: Clustering TREC articles with a threshold of 0.2.

	Unit		With Same Topic		With Similar Topics		With Other Topics	
Topic	Clusters	Percent	Clusters	Percent	Clusters	Percent	Clusters	Percent
113	180	36.7%	78	45.9%	81	4.7%	29	4.7%
114	54	17.4%	45	62.1%	81	15.4%	28	4.8%
121	302	57.3%	73	33.0%	28	7.0%	13	2.7%
122	51	31.3%	18	33.1%	25	33.7%	2	1.8%
123	168	28.2%	87	51.7%	38	17.3%	13	2.5%
124	112	35.1%	22	27.3%	52	35.7%	5	1.9%

Number of clusters: 1300

4.3.3 Discussion of results

One thing that is easily discernible from these tables is that the similarity function for synset weight vectors works as expected. As the threshold increases, the number of vectors clustered from different topics decreases. At the 0.2 level, the majority of the documents are either in clusters by themselves or with documents relevant to the same topic.

The number of clusters produced is quite high in all cases. This is to be expected, since documents that may be relevant to a particular topic may not be entirely related to each other, leading to a low similarity score. In fact, we begin to see that the clusters divide up the set of all documents relevant to a topic into subsets centered around a particular subject. For example, using a threshold of 0.2, two clusters are formed containing articles about high-definition television. All of these articles are classified as relevant to topic 113, but they do form a sub-topic that is recovered during clustering.

This experiment has also shown that it is possible to link articles across newspapers, as many of the clusters contained articles from the AP, WSJ, and SJM corpora. It is also worth noting that the methodology seemed to work just as well for the ZF corpus, which contains some magazine-style articles as long as 77 paragraphs.

5 Evaluation

Clearly, there is a need for evaluation when building systems such as the one that we have built, and so we intend to perform an evaluation of our hypertext generation methodology. We have decided to use a question-answering task because this is the type of task that is best done using the browsing methodology that hypertext embodies. We explicitly make no claims that our hypertext would be useful for all information access tasks, as this is clearly not the case. Our evaluation system will require a standard IR system to retrieve articles to be used as starting points for browsing.

Each subject will be provided with two questions to answer. These two questions will be drawn from the TREC topics that were discussed in sections 4.3. We have decided to use TREC queries because the difficult work of determining which articles are relevant to the query has already been done. The test database will consist of approximately 30,000 articles from the TREC database.

Essentially, we will be comparing our techniques for creating links within and between documents to other possible techniques. For example, we will compare how the subjects perform when the inter-article links are built using our technique or when they are built using an IR system, in this case the Managing Gigabytes system (Witten *et al.*, 1994).

6 Conclusions and future work

One of the advantages of Allan's work (1995) is that the links between portions of two texts can be given a type that reflects what sort of link is about to be followed. We currently have no method for producing such typed links, but it may be the case that the relations between words from WordNet can be used to determine the type of some links.

It is still not clear how much of our methodology depends on the structure of the newspaper articles that we are processing. Does this standard structure enhance our hypertext linking capabilities, or would the method perform equally well, given any well-written text to work with? We intend to see how well the method performs on other types of texts, possibly changing our methodology to cope with the loss of some structure.

While other automatic hypertext generation methodologies have been proposed, many of them rely on term repetition to build links within and between documents. If there is no term repetition, there are no links. This is especially a problem when attempting to build intra-document links in shorter documents when an author may have been striving to avoid using the same word again and again and so chose a related word. We avoid this problem by using lexical chains, which collect words on the basis of their semantic similarity. Our results to date have shown promise for the methodology, and work is continuing.

Acknowledgements

The author wishes to thank Graeme Hirst and Lisa Chislett for their comments on earlier versions of this paper. Thanks also to the *Globe and Mail* for providing the test data and to the anonymous reviewers for their helpful comments. Funding for this work was provided by NSERC and the ITRC.

References

(Allan, 1995) James Allan. *Automatic hypertext construction*. PhD thesis, Cornell University, 1995.

(Associated Press, 1992) Associated Press. U.S. dollar falls in slow day. *The Globe and Mail*, page B16, October 13 1992.

(Beckwith *et al.*, 1991) Richard Beckwith, Christiane Fellbaum, Derek Gross, and George A. Miller. WordNet: A lexical database organized on psycholinguistic principles. In Uri Zernik, editor, *Lexical acquisition: Exploiting on-line resources to build a lexicon*, pages 211–231. Lawrence Erlbaum Associates, 1991.

(Bernstein, 1990) Mark Bernstein. An apprentice that discovers hypertext links. In N. Streitz, A. Rizk, and J. André, editors, *Hypertext: Concepts, systems and applications: Proceedings of the European conference on hypertext*, pages 212–223. Cambridge University Press, 1990.

(Ellis *et al.*, 1994a) David Ellis, Jonathan Furner-Hines, and Peter Willett. On the creation of hypertext links in full-text documents: Measurement of inter-linker consistency. *The Journal of Documentation*, **50**(2):67–98, 1994.

(Ellis *et al.*, 1994b) David Ellis, Jonathan Furner-Hines, and Peter 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, April 1994.

(Ellis *et al.*, 1996) David Ellis, Jonathan Furner, and Peter Willett. On the creation of hypertext links in full-text documents: Measurement of retrieval effectiveness. *Journal of the American Society for Information Science*, **47**(4):287–300, 1996.

(Halliday and Hasan, 1976) M.A.K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976.

(Harman, 1994) Donna Harman. Overview of the third Text Retrieval Conference (TREC-3). In *Proceedings of the third Text Retrieval Conference*, November 1994.

(Morris and Hirst, 1991) Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, **17**(1):21–48, 1991.

(Outing, 1996) Steve Outing. Newspapers online: The latest statistics. *Editor and Publisher Interactive [Online]*, May 13 1996. Available at http://www.mediainfo.com:4900/ephome/news/newshtm/stop/stop513.htm.

(Salton, 1989) Gerard Salton. *Automatic text processing*. Addison-Wesley, 1989.

(St-Onge, 1995) David St-Onge. *Detecting and correcting malapropisms with lexical chains*. Master's thesis, University of Toronto. Published as technical report CSRI-319, 1995.

(Westland, 1991) J. Christopher Westland. Economic constraints in hypertext. *Journal of the American Society for Information Science*, **42**(3):178–184, 1991.

(Witten *et al.*, 1994) Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and indexing documents and images*. Van Nostrand Reinhold, 1994.