Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2005-034 May 19, 2005
AIM-2005-017

Lexical Chains and Sliding Locality Windows in Content-based Text Similarity Detection

Thade Nahnsen, Ozlem Uzuner, Boris Katz

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Abstract

We present a system to determine content similarity of documents. More specifically, our goal is to identify book chapters that are translations of the same original chapter; this task requires identification of not only the different topics in the documents but also the particular flow of these topics. We experiment with different representations employing n-grams of lexical chains and test these representations on a corpus of approximately 1000 chapters gathered from books with multiple parallel translations. Our representations include the cosine similarity of attribute vectors of n-grams of lexical chains, the cosine similarity of tf*idf-weighted keywords, and the cosine similarity of unweighted lexical chains (unigrams of lexical chains) as well as multiplicative combinations of the similarity measures produced by these approaches. Our results identify fourgrams of unordered lexical chains as a particularly useful representation for text similarity evaluation.

1 Introduction

This paper addresses the challenge of determining content similarity in the domain of literary novels. Literary novels provide a partially restricted domain in which authors mostly adhere to the standards of written English but also use colloquial expressions, e.g., in dialogs, monologues, and to convey a specific atmosphere. There has been considerable work on determining the subject of texts and on understanding the semantic relations within a sentence. Our work differs from these in its goal: we aim to determine content similarity between chapters of books. Each chapter can contain more than one topic, and identifying chapters that are similar in content requires resolving exact content matches rather than identifying similarities in dominant topics.

Our approach to identifying content similarity between book chapters relies on a handcrafted knowledge base, WordNet (Miller, 1991; Fellbaum, 1998), to identify semantically related words that contribute to lexical chains and help evaluate content similarity.

2 Related Work

Lexical Chains are sets of lexical items, which are conceptually related to each other. Semantic resources such as WordNet (Miller, 1991; Fellbaum, 1998;) provide information about semantic relations between individual lexemes and can be used to determine conceptual similarity.

Conceptual similarity of lexical items has previously been used in evaluating cohesion, for example, by Halliday and Hasan (1976) who studied linguistic phenomena such as anaphora, cataphora, hyponymy, and synonymy of lexemes (Halliday and Hasan, 1989). Morris and Hirst (1991) suggested creating lexical chains from conceptually similar lexical items in order to determine the discourse structure of texts. Employing WordNet as a semantic knowledge source for building lexical chains was proposed by Hirst and St-Onge (1998).

Barzilay and Elhadad (1999) used lexical chains to create intermediate representations for text summarization; they identified important sentences in a document by retrieving strong chains, employing the hyponymy and synonymy phenomena identified by Halliday and Hasan (1976, 1989). Silber and McCoy (2002) extended the work of Barzilay and Elhadad (1999); they developed an algorithm that is linear in time and space for efficient identification of lexical chains in large documents. Their algorithm first created a text representation in the form of metachains, i.e., chains that capture all possible lexical chains in the document. After creating the metachains, they used a scoring algorithm to identify the lexical chains that are most relevant to the document, eliminated from the metachains unnecessary overhead information, and generated the lexical chains representing the document.

Galley and McKeown (2003) identified lexical chains after performing word sense disambiguation (WSD). Their implementation of lexical chains assumed “one sense per discourse”. This algorithm first constructed a representation of all possible interpretations of the text (similar to metachains in the case of Silber and McCoy), disambiguated all words, and then constructed the lexical chains.

While Gale, Church, and Yarowsky (1992) claim that the "one sense per discourse" constraint frequently holds, Krovetz (1999) showed using semantic relations in WordNet that, polysemous words occur in more than one sense per discourse much more frequently than determined by Gale, Church and Yarowsky (1992). Thus, our algorithm for building lexical chains follows the work by Morris and Hirst (1991), Barzilay and Elhadad (1999), and Silber and McCoy (2002), which do not make this assumption.

N-gram based language models, i.e., models that divide text into *n* word (or *n* character) strings, are frequently used in natural language processing, e.g., Hull et. al, 1982; Church, 1995. In plagiarism detection, the overlap of n-grams between two documents has been used to determine whether two documents plagiarize each other (Lyon, Malcolm, Dickerson, 2001).

In general, n-grams capture local relations. In our case, they help capture local relations between lexical chains and identify local relationships between concepts. As a result, they enable identification of interdependencies between concepts and provide more accurate models of content than approaches that rely merely on individual concepts.

Three main streams of research in content similarity detection are: 1) shallow, statistical analysis of documents (Marcu, 1997), 2) analysis of rhetorical relations in texts (Mann, Thompson, 1988), and 3) deep, syntactic analysis (Uzuner, Davis, Katz, 2004). Shallow methods do not include much linguistic information and therefore provide a very rough model of content while approaches that use syntactic analysis generally require significant computation. Our approach strikes a compromise between the two extremes; it uses the linguistic knowledge provided in WordNet as a way of making use of low cost linguistic information for building lexical chains that can detect content similarity.

In this paper, we evaluate several alternative lexical chain-based systems that capture content similarity. We evaluate these systems on a corpus of parallel translations which includes a total of more than 1000 chapters (475 unique chapters with 2 or 3 translations each).

3 Lexical Chains in Content Similarity Detection

3.1 Corpus

The experiments presented in this paper were carried out on a corpus consisting of chapters from translations of four books. This corpus included 47 chapters from each of two translations of *20,000 Leagues under the Sea* (Verne), 35 chapters from each of 3 translations of *Madame Bovary* (Flaubert), 28 chapters from each of two translations of *the Kreutzer Sonata* (Tolstoy), and 365 chapters from each of two translations of *War and Peace* (Tolstoy).

These books cover a variety of different topics; the topics covered in each of the chapters range from romance to diplomacy, to sea life, and to war. Many of the chapters from each book deal with similar topics, although they do not correspond to translations of the same original chapter. Therefore, fine-grained content analysis is required to identify chapters that are similar in content, i.e., that are derived from the same original and therefore cover the same topics and the same overall content.

3.2 Computing Lexical Chains

Our approach to calculating lexical chains uses the nouns, verbs, and adjectives present in WordNetV2.0. In the first step, we extract nouns, verbs, and adjectives from each chapter in the corpus and represent each chapter as a set of these word instances $\{I_1, \ldots, I_n\}$. Each instance of each of these words has a set of possible interpretations, I_N, in WordNet. These interpretations are either the synsets or the hypernyms of the instance. Given these interpretations, we apply the algorithm by Silber and McCoy (2002) to automatically disambiguate nouns, verbs, and adjectives, i.e., to select the correct interpretation, for the instance. In pseudocode, this algorithm works as follows:

Step 1
For each word instance
    For each sense of the word instance
        Compute all scored metachains
Step 2
For each word instance
    For each metachain the word belongs to
        Keep word instance in the metachain to

which it contributes most
Update the scores of each other metachain

"The furniture in the kitchen seems beautiful, but the bathroom seems untidy."

Gloss of MetaChains/Lexical Chains:

{03281101} (furniture) furnishings that make a room or other area ready for occupancy
{03485400} (kitchen) a room equipped for preparing meals
{02071636}(seem) give a certain impression or have a certain outward aspect
{02072871} (seem) seem to be true, probable, or apparent
{02665354} (seem) appear to exist
{02135322} (seem) appear to one's own mind or opinion
{00218842} (beautiful) delighting the senses or exciting intellectual or emotional admiration
{00074044} (beautiful) aesthetically pleasing
{01740540} (beautiful) (of weather) highly enjoyable
{02709425} (bathroom) a room (as in a residence) containing a bath or shower and usually a washbasin and toilet
{04274300} (bathroom) a room equipped with toilet facilities
{02336718} (untidy) not neat and tidy
{00942215} (untidy) careless and slovenly
{03951013} (room) an area within a building enclosed by walls and floor and ceiling

Possible interpretations for instances:

$I_{furniture}$ = {**03281101**}
$I_{kitchen}$ = {03485400, **03951013**}
I_{seems} = {**02071636**, 02072871, 02665354, 02135322}
$I_{beautiful}$ = {**00218842**, 00074044, 01740540}
$I_{bathroom}$ = {02709425, 04274300, **03951013**}
I_{seems} = {**02071636**, 02072871, 02665354, 02135322}
I_{untidy} = {**02336718**, 00942215}

Identified metachains and lexical chains. Lexical chains are shown in bold l

03281101 = {**furniture**}
03485400 = {kitchen}
02071636 = {**seems**, **seems**}
02072871 = {seems, seems}
02665354 = {seems, seems}
02135322 = {seems, seems}
00218842 = {**beautiful**}
00074044 = {beautiful}
01740540 = {beautiful}
02709425 = {**bathroom**}
04274300 = {bathroom}
02336718 = {**untidy**}
00942215 = {untidy}
03951013 = {**kitchen**, **bathroom**}

Figure 1: Example process for determining lexical chains

Silber and McCoy's mechanism for computing the contribution of a word to a given chain considers 1) the semantic relation between the synsets of the words that are members of the same metachain, and 2) the distance between their respective instances in the discourse. Similarly, our approach uses these two parameters, with minor modifications to accommodate for the differences in genre. Silber and McCoy measure distance in terms of paragraphs on prose text; we measure distance in terms of sentences in order to handle both dialogue and prose text– a different approach would be to determine the genre, i.e., dialog or prose, of a section beforehand, to employ the scheme by Silber and McCoy for prose text, and to determine a new scheme for dialog text.

Following Silber and McCoy, we allow different types of conceptual relations to contribute differently to the lexical chain, i.e., the contribution of each word to the lexical chain is dependent on its semantic relation to the chain. Table 1 summarizes the scheme used to compute the contributions of words to metachains in our system. Concepts that are dominant in the text segment are thus identified and each word is represented by only the synset that best fits its local context; this synset does not have to be a synset of the word itself. The synset that best fits a word's local context becomes that word's lexical chain. Figure 2 shows the results of this algorithm and the interpretation, S, found for each word instance, I, that can be used to represent each chapter, C, where C = $\{S_1, \ldots, S_m\}$. Figure 1 illustrates the process of determining lexical chains. The cases of "kitchen" and "bathroom" show that words can belong not only to lexical chains of their own synsets, but also to lexical chains of their hyponyms. In the example shown, "kitchen" and "bathroom" both have hyponymy relations with the lexical chain representing the concept "room" and hence this chain has higher score than the individual chains of synsets of the terms "kitchen" and "bathroom".

Lexical-semantic relation	Distance <= 6 sentences	Distance > 6 sentences
Same word	1	0
Hyponym	0.5	0
Hypernym	0.5	0
Sibling	0.2	0

Table 1: Contribution to lexical chains

3.3 Determining the locality window

After computing the lexical chains, we created a representation for text by substituting the correct lexical chain for each noun, verb, and adjective in each document. We omitted the remaining parts of speech from the documents (see Figure 2 for sample intermediate representation). We obtained ordered and unordered n-grams of lexical chains from this intermediate representation.

Ordered n-grams consist of *n* consecutive lexical chains extracted from text. These ordered n-grams preserve the original order of the lexical chains in the text. Corresponding unordered n-grams disregard this order. The resulting text representation is $T = \{gram_1, gram_2, \ldots, gram_n\}$, where $gram_i = [lc_j, \ldots, lc_m]$, where $lc_i \in \{I_1, \ldots, I_k\}$ (the chains that represent Chapter C). The elements in $gram_i$ may be sorted or unsorted, depending on the selected method.

N-grams are extracted from text using sliding locality windows and provide what we call "attribute vectors". The attribute vector for ordered n-grams has the form

$C = \{(e_1,\ldots, e_n), (e_2,\ldots, e_{n+1}), \ldots, (e_{m-n},\ldots, e_m)\}$

where $(e_1,\ldots, e_n)$ is an ordered n-gram and e_m is the last lexical chain element in the chapter.

For unordered n-grams, the attribute vector has the form

$C = \{sort[(e_1,\ldots, e_n)], sort[(e_2,\ldots, e_{n+1})], \ldots, sort[(e_{m-n},\ldots, e_m)]\}$ where sort[…] indicates that the order of the elements in the n-gram is disregarded. Figure 3 shows a sample n-gram attribute vector $\{gram_1, \ldots, gram_n\}$ of chapter C.

3.4 Determining Similarity

We designed and implemented three general methods for text similarity based on the n-grams of lexical chains. We used these methods to determine pair-wise similarity between all of the chapters in the corpus. Our three methods differed in the degree to which they preserved the original order of the lexical chains in a text in order to capture the effect of text reordering that may take place during paraphrasing, be it for creating derivative works or for plagiarism. They also differed in whether they used only the cosine similarity of attribute vectors of n-grams or whether they also included tf*idf-weighted keywords in the representation.

Original document (underlined words are represented with lexical chains):

The year 1111 was signalized by a remarkable incident, a mysterious and puzzling phenomenon, which doubtless no one has yet forgotten, not to mention rumors which agitated the maritime population and excited the public mind, even in the interior of continents, seafaring men were particularly excited.

Intermediate representation (lexical chains):

14343019 13161930 6373972 1627764 6858832 898635 505730 29881 12489711 1558297 6347853 6780062 147724 2753308 7683412 13236825 1799910 5291080 8063710 8677230 1333878 7715015 13161930 13236825

Figure 2: Intermediate representation after identifying lexical chains, eliminating words that are not nouns, verbs or adjectives, and after performing WSD

For the first two methods, we applied the cosine similarity to the attribute vectors of ordered and unordered n-grams (sliding locality windows of width *n*) of lexical chains. For both cases, we varied the width of the sliding locality windows from two to five elements. We created the third measure of similarity through a multiplicative combination of the similarity scores obtained through the baseline of tf*idf-weighted keywords and attribute vectors of n-grams of unordered lexical chains.

14343019 13161930
13161930 6373972
6373972 1627764
1627764 6858832
6858832 898635
898635 505730
505730 29881
29881 12489711
12489711 1558297

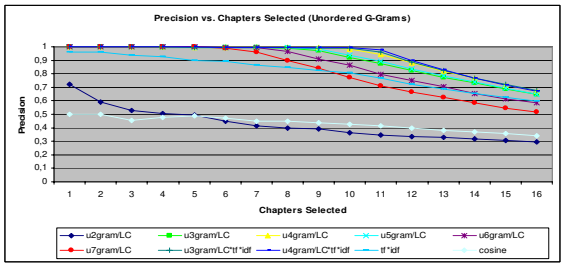
Figure 3: Example of attribute vector (bigram vector) used for calculating similarity (excerpt)

4 Evaluation

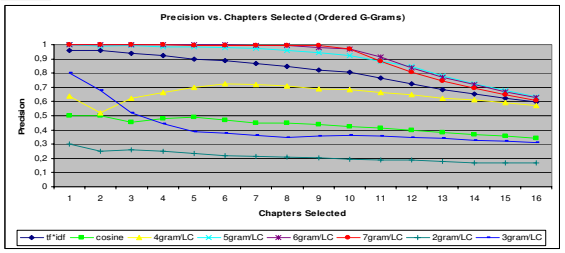
We take a nearest-neighbors approach to evaluating similarity. We use the cosine similarity as the distance metric, compute the cosine of the angles between the vectors of pairs of documents in the corpus, and rank the pairs based on this score. Using this ranking, we identify the top *n* most similar

pairs (also referred to as "selection level of *n*") and consider them to be similar in content.

We calculated similarity between pairs of documents in several different ways, evaluated these approaches with the standard information retrieval measures, i.e. precision, recall, and f-measure, and compared our results with two baselines. The first baseline measured the similarity of documents with tf*idf-weighted keywords; the second used the cosine of unweighted lexical chains (unigrams of lexical chains).



(a) Precision: Unordered N-Grams versus the baselines



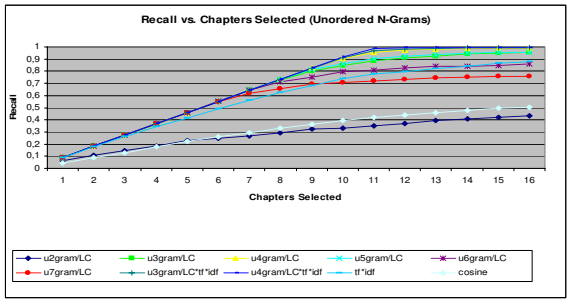
(b) Precision: Ordered N-Grams versus the baselines

Figure 4: Precision[1]

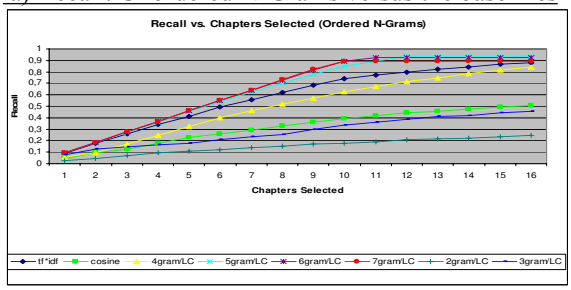
The corpus of parallel translations provides data that can be used as ground truth for content similarity. In particular, corresponding chapters from different translations of the same original title are considered similar in content, i.e., chapter 1 of translation 1 of *Madame Bovary* is similar in content to chapter 1 of translation 2 of *Madame Bovary*. These corresponding chapters are derived from the same original chapter, and therefore include the same content. The total number of chapter pairs in this corpus is ca. 1,000,000 pairs. Of these, 1,080 (475 unique chapters with 2 or 3 translations each) are similar.

Figures 4, 5, and 6 show the precision, recall, and f-measure of different methods for measuring similarity between pairs of chapters using ordered lexical chains, unordered lexical chains, and baselines. These graphs present the results when different numbers of chapters are considered similar (i.e., different selection levels). At selection level 1,100, for example, the top 1,100 most similar pairs in the corpus are considered similar in content and the rest are considered dissimilar.

Figure 6 indicates that the similarity evaluation methods used in this paper can be divided into four groups: i) methods with f-measure less than 0.418; ii) methods with f-measure between 0.6 and 0.826; iii) methods with f-measure between 0.85 and 0.9; and iv) methods with f-measure above 0.92.



a) Recall: Unordered N-Grams versus the baselines



(b) Recall: Ordered N-Grams versus the baselines

Figure 5: Recall[1]

In our experiments, three similarity measures outperformed all others (but gave almost identical performance to each other). These were the measures in group (iv) which contained 1) a combination of the cosine of trigrams of unordered lexical chains[2] with the cosine of tf*idf-weighted keywords; 2) a combination of the cosine of fourgrams of unordered lexical chains with the cosine of tf*idf-weighted keywords; and 3) fourgrams of

[1] **n**gram/LC – order-maintaining ngrams of lexical chains are used as the attribute vector for similarity detection
u**n**gram/LC – order-disregarding ngrams of lexical chains are used as attribute vector for similarity detection
tf*idf – tf*idf weighted words are used as attribute vector for similarity detection
u**n**gram/LC*tf*idf – multiplicative combination of u**n**gram/LC and tf*idf
cosine – words are used as attribute vector for similarity detection

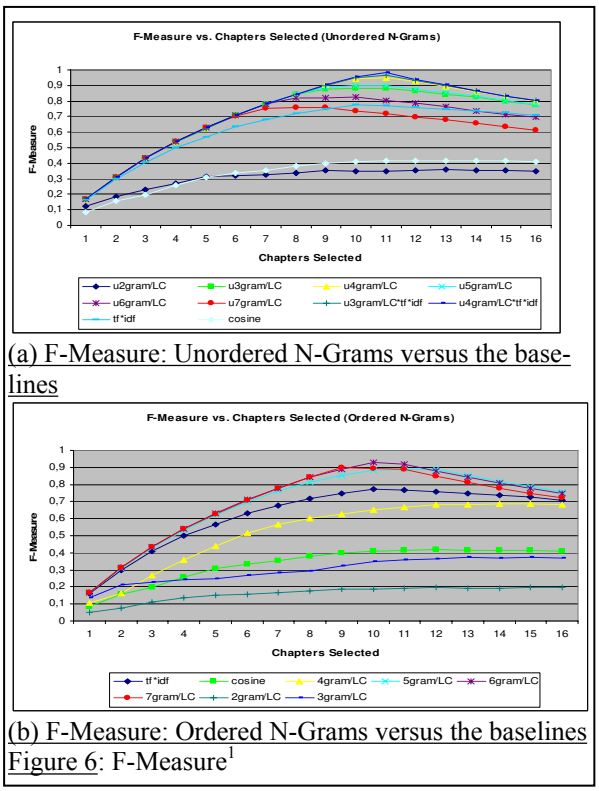
[2] We refer to *n-grams of (un)ordered lexical chains* also as *(un)ordered n-grams*.

unordered lexical chains. The peak f-measure at selection level of 1,100 chapter pairs was 0.989. Chi squared tests performed on the f-measures (when the top 1,100 pairs were considered similar) were significant at p=0.001 for the three measures in group (iv) compared to all others; the significance was p=0.01 for sixgrams of ordered lexical chains. P-values for the significance tests on the differences in the performance of the different methods for a range of selection levels (in the interval 800 to 1,300) are shown in Table 2. The difference of the three highest ranking lexical chain n-grams (from the baseline and from each other) is significant at a confidence level of at least 0.05 in the interval 900 to 1,300 (see Table 2). Similar significance levels were observed for the difference of group (iii) from the baseline in this interval.

N-grams of ordered lexical chains of length greater than four significantly outperform the baseline (in f-measure) at p = 0.001 while n-grams of ordered lexical chains of length less than or equal four are significantly outperformed by the baseline at the same p. A similar observation cannot be made for the n-grams of unordered lexical chains; n-grams of unordered lexical chains of length five or more significantly outperform the baseline provided by tf*idf weighted keywords when the top 700 chapter pairs are considered similar, but are significantly outperformed by the baseline when 1,300 chapter pairs are considered similar. Hence, unordered n-grams with n >= 5 indicate similarity effectively for the most similar pairs but their performance degrades more rapidly than the tf*idf baseline as the cut-off for the number of chapter pairs that are considered similar increases.

	800	900	1,000	1,100	1,200	1,300
U4/lc vs 6/lc	0.1	0.01	0.1	0.01	0.01	0.01
u4/lc vs u5/lc	0	0.01	0.001	0.001	0.01	0.05
u4/lc vs 5/lc	0.001	0.001	0.001	0.001	0.025	0.05
U4/lc vs tf*idf	0.001	0.001	0.001	0.001	0.001	0.001

Table 2: Chi squared significance: p values



(a) F-Measure: Unordered N-Grams versus the baselines

(b) F-Measure: Ordered N-Grams versus the baselines

Figure 6: F-Measure[1]

Figures 6a and 6b show that after the cut-off point of 1,100 chapter pairs, the performance of all algorithms decline and approach similar values. This is expected partly because of the evaluation method we have chosen: although the cut-off for similarity judgement can be increased, the number of chapters that are in fact similar does not change and therefore at high cut-off values many dissimilar pairs are considered similar, leading to degradation in performance

Figures 6a and 6b shows that some of the lexical chain representations do not outperform the tf*idf-weighted baseline. A comparison of Figures 6a and 6b shows that, for n<5, n-grams of ordered lexical chains perform worse than the n-grams of unordered lexical chains. This indicates that between different translations of the same book the order of chains changes significantly, but that the chains within contiguous regions (locality windows) of the texts remain similar. This fact makes longer n-grams a good indicator of similarity: while bigrams can be common between documents that do not share content, n-grams of five or more elements are common mostly to documents that share content, i.e., parallel translations of the same original.

The nearly identical performance of the top 10 measures in precision, recall, and f-measure at selection level below 700 indicates that the highest ranked chapters have large n-grams in common, while the lower ranked chapters predominantly have shorter n-grams in common because the sharp degradation in performance for selection levels above 1,100 does not occur in the case of shorter n-grams (see Figure 6).

The difference in the performance of n-grams of ordered and unordered lexical chains on this corpus is expected. Translation of documents is a subjective task; translators can translate content in their own style and add their own artistic value to the work. During this process, translators may reorder some elements. However, our findings show that most such arrangements are local, i.e., within a four chain window. Beyond this window size, our results show that unordered n-grams perform worse than ordered n-grams (compare Figures 6a and 6b). Conversely, within the four chain window, unordered n-grams outperform the ordered n-grams. At n = 5, the two approaches yield nearly identical results.

Figure 6a shows that the multiplicative combination of similarity judgements on n-grams of lexical chains with similarity measures obtained from tf*idf-weighted keywords gives slightly better performance than n-grams of lexical chains and tf*idf-weighted keywords alone. In the case of u4-gram/LC and u4-gram/LC*tf*idf, the difference in the perforrnances of the two measurements is not significant. u4-gram/LC gives almost perfect performance, and tf*idf-weighted keywords do not really contribute much to performance.

5 Future Work

One of the problems of the similarity detection schemes presented in this paper is the uniform weighting of all n-grams that appear in documents. For example, the n-gram "be it as it has been" in lexical chain form corresponds to synsets for the words be, has and been. The trigram of these lexical chains does not convey significant meaning. On the other hand, the n-gram "the lawyer signed the heritage" is converted into the trigram of lexical chains of lawyer, signed, and heritage. This trigram is more meaningful than the trigram "be has been" but in our scheme, both trigrams will have the same weight. As a result, two documents that share the trigram "be has been" will look as similar as two documents that share "lawyer signed heritage." This problem can be addressed in two possible ways: using a 'stop word' list to filter such expressions completely or giving different weights to n-grams (of lexical chains) based on the number of their occurrences in the corpus.

Future work will also investigate whether similarity detection based on lexical chains can be generalized to other document genres, such as news corpora.

6 Conclusion

We presented a system for content similarity detection that employs lexical chains and sliding locality windows. This system evaluates similarity between pairs of chapters of novels using cosine similarity on n-grams of lexical chains and tf*idf weighted keywords. While domain-dependent approaches to content similarity detection focus on deep semantic analysis, our approach identifies shallow semantics using information provided in WordNet and applies statistical methods (mostly used in domain-independent approaches) to determine similarity.

This method for evaluation of similarity extends previous work on lexical chains to content similarity detection. The results indicate that lexical chains are effective for detecting content similarity between pairs of chapters corresponding to the same original in a corpus of parallel translations; they outperform keyword-based approaches, i.e., tf*idf-weighted keywords and unweighted keywords, in all of precision, recall, and f-measure.

References

Barzilay, R., Elhadad, M. 1999. Using lexical chains for text summarization. In: Inderjeet Mani and Mark T. Maybury, editors, *Advances in AutomaticText Summarization*, pp. 111–121. Cambridge/MA, London/England: MIT Press.

Church, K. W. 1995. Ngrams. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.

Fellbaum, C, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Gale, W., Church, K., Yarowsky, D. 1992. A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities*, 26, pp. 415 – 439.

Galley, M., McKeown, K. 2003. Improving word sense disambiguation in lexical chaining. In *proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico.

Halliday, M.A.K. and Hasan, R. 1976, *Cohesion in English*. Longman, London.

Halliday, M.A.K. and Hasan, R. 1989, *Language, context, and text*. Oxford University Press, Oxford, UK.

Hirst, G., St-Onge, D.. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database and some of its applications*, pp. 305–332. Cambridge, MA: The MIT Press.

Hull, J. J. and Srihari, S. N. 1982. Experiments in text recognition with binary n-gram and Viterbi algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4(5):520-530.

Krovetz, R. 1998. More than one sense per discourse. Research Memorandum. NEC Princeton NJ Labs.

Lyon, C., Malcolm, J. and Dickerson, B. 2001. Detecting Short Passages of Similar Text in Large Document Collections, In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp.118-125.

Mann, W. C. and Thompson, S.A. Rhetorical Structure Theory: A theory of text organization. In L. Polanyi (ed.) *The Structure of Discourse*, Ablex, 1988.

Marcu, D. 1997. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts (Ph.D. dissertation)*. Univ. of Toronto.

Miller G., Beckwith R., Felbaum C., Gross D., and Miller K. 1990. Introduction to WordNet: An online lexical database. *J. Lexicography*, 3(4), pp.235-244.

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

Silber, G. and McCoy, K. 2002. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4), pp. 487–496.

Uzuner, O., Davis, R., Katz, B. 2004. Using Empirical Methods for Evaluating Expression and Content Similarity. In: *The Proceedings of the 37th Hawaiian International Conference on System Sciences* (HICSS-37), Hawaii, HI, January 5-8, 2004. IEEE Computer Society.