Lexical Chains Using Distributional Measures of Concept Distance

Meghana Marathe and Graeme Hirst

University of Toronto, Toronto, ON, M5S3G4
mm@cs.toronto.edu, gh@cs.toronto.edu

Abstract. In practice, lexical chains are typically built using term reiteration or resource-based measures of semantic distance. The former approach misses out on a significant portion of the inherent semantic information in a text, while the latter suffers from the limitations of the linguistic resource it depends upon.

In this paper, chains are constructed using the framework of distributional measures of concept distance, which combines the advantages of resource-based and distributional measures of semantic distance. These chains were evaluated by applying them to the task of text segmentation, where they performed as well as or better than state-of-the-art methods.

1 Introduction

Lexical chains are sequences of semantically related words in a text. A word is added to an existing chain only if it is related to one or more of the words already in the chain by a *cohesive* relation. In practice, the cohesion between two words is approximated either by term reiteration or by the *semantic distance* between them. Methods that restrict lexical cohesion to reiteration consider two terms to be related only if they are instances of the same word. Hence, these methods miss out on a significant portion of the semantic information inherent to a text.

Semantic distance is typically computed using linguistic resource-based measures or measures of distributional similarity, both of which have inherent disadvantages. This motivates the need for a hybrid that incorporates the advantages of both these methods. Mohammad and Hirst (2006) proposed *distributional measures of concept distance (DMCDs)* that combine distributional co-occurrence information with semantic information from a lexicographic resource, such as a thesaurus. These measures were shown to outperform traditional distributional measures on the tasks of correcting real-word spelling errors, and ranking word pairs in order of semantic distance. In this work, we build lexical chains using Mohammad and Hirst's framework of distributional measures of concept distance. The chains are evaluated by applying them to the task of text segmentation.

Text segmentation is the task of dividing a text document into cohesive units or segments by topic (Hollingsworth 2008). In particular, we focus upon *linear* segmentation, in which segments are not further subdivided; as opposed to *hierarchical* segmentation, where each unit may in turn be divided into sub-units.

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 291–302, 2010.

Morris and Hirst (1991) were the first to suggest using lexical chains for text segmentation, which has since become a standard application of lexical chains. Since lexical chains consist of semantically related words, each chain corresponds to a theme or topic (or a set thereof) in the text. As a result, lexical chains provide three useful cues, namely:

- A significant number of chains beginning at a point in text probably indicates the emergence of some new topic(s).
- A significant number of chains ending at a point in text probably means that certain topics are not discussed henceforth in the text.
- Points where the number of chains beginning or ending is not significant probably represent a continuation in the discussion of some topic(s).

Our hypothesis is that these cues help detect positions at which there are changes or shifts in topic, representing segment boundaries.

2 Background

This section provides a review of previous work in lexical chaining and text segmentation, and provides the motivation for the proposed method.

2.1 Lexical Chains

Halliday and Hasan (1976) laid the foundation for lexical chains, when they suggested relating words of a text back to the first word to which they are cohesively "tied". They also specified five types of lexical cohesion based on the dependency relationship between the words. However, they did not consider exploiting the transitivity of these relationships, nor did they discuss computational methods for finding lexical chains.

Morris and Hirst (1991) were the first to suggest computational means of building lexical chains. They used the hierarchical structure of *Roget's International Thesaurus, 4th Edition (1977)* to find lexical relationships between words. Based on their analysis of five texts, Morris and Hirst concluded that lexical chains computed by their algorithm correspond closely to the intentional structure[1] of that text produced from the structural analysis method of Grosz and Sidner (1986). Unfortunately, no online copy of the thesaurus was available to Morris and Hirst, so the algorithm was worked out by hand, preventing extensive tests.

There have since been several attempts at constructing lexical chains using WordNet (Fellbaum 1998), a large lexical database for English. The structure of WordNet being quite different from that of *Roget's*, researchers proposed new notions of semantic relatedness. Hirst and St-Onge (1998), for instance, classifed WordNet synset relations into upward, downward, and horizontal directions. For a given pair of words, the connections between some synset of one word and some synset of the other and the directions of these connections determine how related the words are.

[1] Intentional structure is based on the idea that every discourse has an overall purpose; and that every discourse segment has a purpose, specifying how it contributes to the overall purpose.

Stokes et al. (2004) proposed the use of lexical chaining as a means of segmenting news stories. They experimented with synonymy, specialization, and part-whole relationships from WordNet; and statistical word association as indicators of lexical cohesion for building chains. Even so, they concluded that optimal performance was achieved when only noun repetition patterns were examined during boundary detection.

Yang and Powers (2006) employed WordNet together with the Edinburgh Associative Thesaurus (EAT)[2] to build "improved" lexical chains called lexical hubs, for word sense disambiguation (WSD). The EAT consists of an associative network of words, constructed by asking subjects to state the first word they thought of in response to a stimulus word (Kiss et al. 1973). Since WordNet usually restricts itself to paradigmatic relations between words (Fellbaum 1998), the EAT was used to add associative information. This significantly improved results on the WSD task. However it limits the method's scope to resource-rich languages, requiring not only WordNet but also an associative thesaurus.

These methods suffer from WordNet's fine-grainedness, which has been a typical and frequent criticism of WordNet in the literature. Moreover, it is mainly the noun hierarchy of WordNet that has been extensively developed. Hence these methods cannot exploit the information contained in other parts of speech, such as verbs and adjectives.

Strength of a Chain. Lexical chaining algorithms often produce a much larger number of chains than desired for a particular task (Hollingsworth 2008). *Chain strength* is used to select the "best" or most relevant chains out of a given set of chains. Morris and Hirst (1991) first proposed the concept of chain strength, naming three factors that contribute to it: reiteration, density, and length. Reiteration is computed by counting the number of word-tokens of each word-type present in the chain. Chain density is the ratio of the number of words in a chain to the number of content words in the text (Hollingsworth 2008). The length or size of a chain is the number of word-types it contains. Morris and Hirst advocate using a combination of these three factors to compute chain strength.

In practice, chain strength has often been calculated as a weighted sum of the number of occurrences of each word-type in a chain (Barzilay and Elhadad 1997; Hirst and St-Onge 1998; Hollingsworth 2008). The value of a weighting coefficient depends on the kind of lexical relation used to add that term to the chain. It should be noted that this implicitly assumes that the same relation is used to add every occurrence of a word-type to a specific chain.

2.2 Text Segmentation

TextTiling (Hearst 1994, 1997) is widely considered a foundational work in paragraph-level text segmentation. It is an algorithm for partitioning expository texts into coherent multi-paragraph discourse units that reflect the underlying subtopic structure.

Instead of identifying individual subtopics, TextTiling focuses on detecting *subtopic shifts*. It assumes that a significant change in the vocabulary being employed is indicative of a shift from one subtopic to another. It uses term reiteration to detect these shifts. Thus, TextTiling does not depend on any lexical resource or inference mechanisms and can be applied to a variety of natural languages. Unfortunately, the algorithm requires setting several interdependent parameters, with no fixed way of determining the ideal values.

[2] http://www.eat.rl.ac.uk

Okumura and Honda (1994) used a Morris and Hirst style lexical chainer to determine segment boundaries. They hypothesized that when a lexical chain ends, there is a tendency for a segment to end; and when a new chain begins, it might indicate that a new segment has begun. Thus, sentence-gaps with the highest sum of the number of lexical chains beginning or ending at this gap are chosen as segment boundaries.

The authors reported preliminary but encouraging results on five Japanese texts. However they did not present any comparison of the performance of their algorithm with that of a baseline or of another algorithm such as TextTiling.

C99 (Choi 2000) is a domain-independent algorithm for linear text segmentation. A dictionary of word-stem frequencies in vector form is built for each tokenized sentence, and a similarity matrix is generated by computing the cosine similarity between every pair of sentences. Next, each value in the similarity matrix is replaced by its rank in the local region to generate a rank matrix. A text segment k is defined by two sentences, i and j, represented as a square region along the diagonal of the rank matrix. Segments are identified using divisive clustering based on Reynar's maximization algorithm (Reynar 1998).

C99 was shown to outperform TextTiling, DotPlot (Reynar 1998) and Segmenter (Kan et al. 1998) on an artificial test corpus.

2.3 Measures of Semantic Distance

We present a brief overview of the three major classes of methods used to compute semantic distance. For a more complete discussion, please refer to Mohammad and Hirst (2005), and Budanitsky and Hirst (2006).

Resource-based measures are computed using dictionaries, thesauri or wordnets. In a dictionary the semantic distance between two words may, for instance, be defined as the number of common words in the definitions of the two words (Lesk 1986). In a wordnet it could be defined by the amount of information shared by the nodes corresponding to the two words (Lin 1998b). In a thesaurus, semantic distance can be defined in terms of the length of the path between the two words through the category structure or index (Morris and Hirst 1991).

Most of these methods correlate well with human judgements (see Budanitsky and Hirst 2006), but they have several shortcomings due to their dependence on a specific resource, such as the inability to operate across parts of speech (e.g., the semantic distance between a verb and a noun); or the lack of consideration for non-classical relations (e.g., semantic role relation). It also means that they cannot be applied to languages in which those resources do not exist.

Distributional measures treat two words as semantically related if they tend to co-occur with similar contexts. These methods build one distributional profile (DP) per word, consisting of the number of occurrences of that word in various contexts. For example, if the target word is *deluminator* and the corpus contains the sentence '*It was a curious device, his deluminator.*', the method increments the count of occurrences of *deluminator* in the context of *curious* and of *device*.

Measures of distributional similarity typically differ from each other in their notion of context (e.g., a window of n tokens *vs*. a syntactic argument relationship) and the technique used to incorporate co-occurrence information (e.g., conditional probability *vs*. pointwise mutual information).

These measures can be applied across parts of speech and they can also detect non-classical relationships provided these are reflected in the corpus. However, their correlation with human judgements is observed to be fairly low (Weeds 2003), and they require extremely large corpora in order to gather sufficient data. In addition, the methods run into problems with word sense ambiguity because they consider only the surface forms of words and not their meanings.

Hybrid methods aim to combine the advantages of resource-based and distributional methods by using both distributional information and a linguistic resource. Multiple hybrid methods have been proposed, but we discuss here the framework proposed by Mohammad and Hirst (2006).

Their framework of *distributional measures of concept distance (DMCDs)* combines distributional co-occurrence information with the semantic information from a lexicographic resource. Mohammad and Hirst used the categories from the *Macquarie Thesaurus* (Bernard 1986) as a set of coarse-grained word senses or concepts to build a word-category co-occurrence matrix (WCCM) using the sense-annotated *British National Corpus (BNC)*. Cell m_{ij} in the WCCM contained the number of times word i co-occurred (in a window of ±5 words in the corpus) with any of the words listed under category j in the thesaurus. Distributional profiles of concepts (DPCs) could be derived from the WCCM by applying a suitable statistic, such as odds ratio or pointwise mutual information.

A DMCD is defined as any distributional measures in which DPCs of the categories of the target words are used as the context, in place of DPs of the words themselves. A DMCD is thus completely defined by choosing the window size (usually ±5 words), the measure of distributional similarity, and the statistic used to measure the strength of association.

DMCDs were evaluated in comparison with distributional and WordNet-based measures on two tasks: ranking word pairs in order of semantic distance with human norms; and correcting real-world spelling errors. DMCDs outperformed distributional measures on both tasks. They did not perform as well as the best WordNet-based measures in ranking word pairs, but in the spelling correction task, DMCDs beat all WordNet-based measures except that of Jiang and Conrath (1997).

3 Method

In this section we describe the general algorithm used for building lexical chains, the procedure used for segmenting text using chains, and the two variants of the chaining algorithm that were implemented.

3.1 A General Algorithm for Lexical Chains

The lexical chaining algorithm is adapted from the one proposed by Morris and Hirst (1991). It requires the setting of three parameters: an indicator of lexical cohesion I (e.g., a measure of semantic distance); the threshold for adding a word to a chain, $threshold_a$; and the threshold for merging two chains, $threshold_m$. The range of acceptable values for the two thresholds depends upon the range of scores assigned by the method I. The algorithm requires a method $sim\_ww(x,y)$ that computes the lexical cohesion score between words x and y according to indicator I; and expects text in the form of a list of sentences from which punctuation and stop words have been eliminated.

For each word in the text, the algorithm computes the similarity score between that word and each existing chain using equation 1. If there are no existing chains, or if the maximum score obtained is lesser than $threshold_a$, a new chain containing that word is created.

$$sim\_wc(token, chain) = \underset{word \in chain}{average}\,(sim\_ww(token, word)) \tag{1}$$

If there is only one existing chain that obtains the maximum score, the word is added to that chain. If, however, more than one chain obtains the maximum score, these chains become candidates for merging. Similarity scores are computed between each pair of candidate chains using equation 2. If this score exceeds $threshold_m$, the two chains are merged; else the pair is removed from the candidate pairs. This eventually leads to one surviving candidate, to which the word is added. If no chains are merged, the word is added to the first merge candidate.

$$sim\_cc(chain1, chain2) = \underset{w1 \in chain1, w2 \in chain2}{average}\,(sim\_ww(w1, w2)) \tag{2}$$

Once all the words in the text have been processed, the algorithm halts, producing a list of lexical chains. Please refer to algorithm 1 for the pseudocode.

Algorithm 1. Building lexical chains

```
$list\_of\_chains = empty$
**for each** $word$ in $text$ **do**
  $max\_score = \max_{c \in list\_of\_chains} (sim\_wc(word, c))$
  $max\_chain = \operatorname{argmax}_{c \in list\_of\_chains} (sim\_wc(word, c))$
  **if** $list\_of\_chains = empty$ OR $max\_score < threshold_a$ **then**
    Create new chain $c$ containing $word$.
    Add $c$ to $list\_of\_chains$.
  **else if** more than one $max\_chain$ **then**
    Merge chains if needed, adding the word to the resultant chain.
  **else**
    Add $word$ to the chain $max\_chain$.
  **end if**
**end for**
**return** $list\_of\_chains$
```

Interpretation of Parameter Values. Assuming that the indicator I assigns cohesion scores in the range $(0,1)$ (where 0 is assigned to semantically distant pairs of words), increasing $threshold_a$ beyond 0.8 yields highly conservative chains built mainly using term reiteration, whereas decreasing it below 0.5 yields low-coherence chains where the relationship between words is often not clear. Similarly, a high value of $threshold_m$ leads to very infrequent merging; whereas a low value leads to merging of chains that are not very related to each other.

Chain Strength. As noted in section 2.1, chain strength calculations commonly make the assumption that the same relation is used to add every occurrence of a word-type to a specific chain. However, our algorithm uses $sim\_ww(x,y)$ scores to add words to chains, instead of directly perceptible relations. Thus different occurrences of the same word-type may be added to a chain with different scores. Hence, we eliminate weighting from the calculation of chain strength, effectively reducing it to the length or size of the chain.

3.2 Predicting Segment Boundaries

To choose segment boundaries, we use the scoring system described by Okumura and Honda (1994) coupled with a different way of determining the number of boundaries to predict. After chaining, every gap between a pair of consecutive sentences in the text is assigned a score equal to the number of chains beginning and ending at that gap. Boundaries are predicted at gaps whose score exceeds $threshold_{seg}$, computed as a function of the mean gap-score (see procedure 1). The parameter α can either be an absolute value (chosen by tuning it on a development set) or a function of the gap-scores (e.g., variance).

Procedure 1. $predict\_boundaries(text, \alpha)$

```
$score = empty$
$segment\_boundaries = empty$
for each gap $i$ in $text$ do
  $score_i$ = number of chains beginning at $i$ + number of chains ending at $i$
end for
$threshold_{seg} = \underset{gap\ i \in text}{average}(score_i) + \alpha$
for each gap $i$ in $text$ do
  if $score_i \geq threshold_{seg}$ then
    Add $i$ to $segment\_boundaries$.
  end if
end for
return $segment\_boundaries$
```

3.3 Variants

In order to compare performance not only with a state-of-the-art segmentation method, but also with a resource-based semantic measure, we experiment with two variants of the general algorithm. Both use $threshold_a = 0.8$, $threshold_m = 0.5$, and $\alpha = 3$ (tuned using a development set), but differ in their choice of the indicator I:

- *LexChains-Lin*: Here I is Lin's WordNet-based measure (Lin 1998b), implemented in the WordNet::Similarity package (Pedersen et al. 2004). This measure estimates the semantic distance between two words using the amount of information shared by the nodes in WordNet corresponding to these words.

- *LexChains-Saif*: Here I is obtained using Mohammad and Hirst's framework of *distributional measures of concept distance*. In particular, we used Lin's measure of distributional similarity (Lin 1998a) with point-wise mutual information (PMI) as the measure of the strength of association. The Lin-PMI measure was chosen because it consistently performed as well as, if not better than, other DMCDs.

4 Evaluation

This section describes the data and methodology used and the results obtained in the evaluation of the lexical chaining method presented in the earlier section.

4.1 Data Preparation

Creating gold-standard text-segmentation data based on human judgements is very difficult, because intercoder agreement is fairly low (Hearst 1997; Passonneau and Litman 1993). To avoid this problem we used a corpus of research papers, with section- and subsection-boundaries acting as reference segments. Since research papers are written with a view of presenting information in a coherent and structured manner, we believe that the reference segments are a close approximation of gold-standard segments.

The ACL Anthology[3], sponsored by the Association for Computational Linguistics, is the NLP community's research repository. The ACL Anthology Reference Corpus (Bird et al. 2008) is an ongoing effort to provide a standardized reference corpus based on the ACL Anthology. It consists of:

- the source PDF files for articles in the Anthology, as of February 2007;
- raw text for all these articles, extracted automatically from the PDFs using non-OCR based text extraction; and
- metadata for the articles, in the form of BibTeX records.

When we say the text is *"raw"*, we mean that there is no mark-up (to delineate headings or sentences) and that extraction errors (e.g., '...' transcribed as ',Ä¢') have not been corrected. We used 20 raw-text documents from the ACL ARC corpus, manually marking segment boundaries at the end of each section or subsection larger than 2–3 sentences. A simple heuristic-based sentence boundary detection algorithm was used to convert the text into a list of sentences, from which punctuation and stop words were then stripped. This list was given as input to the text segmentation method.

4.2 Methodology

In order to test our hypothesis from section 1, we compare the performance of the two variants of the lexical chaining method on the task of text segmentation with that of JTextTile (Choi 1999), an improved version of TextTiling; and C99; both with default parameter settings.

A segment-boundary is defined by the number of the sentence it occurs after. A *strictly-correct* boundary is one that occurs at the same sentence-gap as a boundary

[3] Available at http://www.aclweb.org/anthology/

in the reference segmentation. A *nearly-correct* boundary is one that is either strictly correct or occurs one gap before or after a boundary in the reference segmentation. We evaluate the segmentation proposed by each method using three sets of measures:

- *Strict precision, strict recall, strict F-score*: Strict precision is the number of strictly-correct proposed segments divided by the total number of segments in the hypothesized segmentation. Strict recall is the number of strictly-correct proposed segments in the hypothesized segmentation divided by the number of segments in the gold-standard segmentation. Strict F-score is the harmonic mean of strict precision and strict recall. For all three measures, the higher the value, the better.
- *Relaxed precision, relaxed recall, relaxed F-score*: These measures are defined the same as their strict counterparts, except for nearly-correct boundaries.
- *Weighted and unweighted WindowDiff*: This metric (Pevzner and Hearst 2002) assigns a score in the range $(0,1)$ to a hypothesized segmentation, where a score of 0 indicates an exact match with the reference segmentation, and a score of 1 indicates that none of the proposed boundaries lie within k sentences of a reference boundary, k being half the average segment length. Weighted *WindowDiff* is defined as follows:

$$\mathit{WindowDiff}(\mathit{ref},\mathit{hyp}) = \frac{1}{N-k}\sum_{i=1}^{N-k}\left|b(\mathit{ref}_i,\mathit{ref}_{i+k}) - b(\mathit{hyp}_i,\mathit{hyp}_{i+k})\right| \quad (3)$$

  Here *ref* is the reference segmentation; *hyp* is the proposed segmentation; $b(p,q)$ is the number of boundaries between positions p and q in the text; and N is the total number of sentences in the text. The i is incremented at each sentence-boundary.

  On the other hand, unweighted *WindowDiff* assigns a penalty of one whenever the absolute difference between the number of boundaries in the reference and hypothesized segmentations (i.e. the value being summed over) exceeds zero.

4.3 Results

The precision, recall, F-score, and *WindowDiff* values for the four methods are reported in Table 1. The best score in each column is rendered in boldface. From the table, it is clear that the two lexical chaining methods, especially LexChains-Saif, outperform the other methods in all metrics.

The difference in the strict and relaxed scores of LexChains-Saif and LexChains-Lin is statistically insignificant[4]. The strict and relaxed scores for LexChains-Saif differ from those of C99 with a confidence interval of 90% and 98% respectively. Similarly, strict precision, and all relaxed scores for LexChains-Saif differ from those of JTextTile, with a confidence interval of 90% and 99% respectively.

While C99 performs nearly as well as LexChains-Saif on weighted *WindowDiff*, on unweighted *WindowDiff* LexChains-Saif outperforms C99 with a confidence interval of 90%, and JTextTile with an interval of 99%.

[4] We used the independent Student's t-test and the Wilcoxon signed-rank test to check whether two sets of samples (scores) arise from statistically different populations.

Table 1. Precision, recall, f-score, and *WindowDiff* values for JTextTile, C99, LexChains-Lin and LexChains-Saif, averaged over 20 documents

Method	Strict			Relaxed			*WindowDiff*	
	Precision	Recall	F-score	Precision	Recall	F-score	Weighted	Unweighted
JTextTile	13.2%	16.4%	14.2%	18.0%	21.9%	19.2%	0.625	0.56
C99	13.0%	14.6%	13.4%	20.4%	23.6%	21.3%	0.595	0.537
LexC-Lin	15.0%	**22.9**%	17.5%	24.7%	**35.8**%	28.3%	0.729	0.515
LexC-Saif	**18.5**%	18.9%	**18.0**%	**29.8**%	31.0%	**29.4**%	**0.577**	**0.463**

5 Conclusion

5.1 Summary of Results

Both variants of the lexical chaining method described significantly outperformed JTextTile (Choi 1999), an improved version of TextTiling (Hearst 1994, 1997). They also outperformed or performed as well as C99 (Choi 2000), a popular domain-independent text-segmentation algorithm. Of the two variants, LexChains-Saif, which used a DMCD, performed better overall than LexChains-Lin, which used Lin's WordNet-based measure (Lin 1998b). This proves our hypothesis.

5.2 Future Work

- *Effects of Genre*: The ACL ARC corpus (Bird et al. 2008) represents the very constrained genre of research papers in the area of Computational Linguistics. It would be interesting to analyze the performance of different measures of semantic distance on a variety of genres; and to investigate the effect(s) of document genre on the evaluation task.
- *Setting Parameter Values*: In this work, $threshold_a$, $threshold_m$ and α, the parameters of the lexical chaining algorithm, were tuned using a small development set. This in itself was difficult because the parameters are interrelated, making it hard to isolate their effects. It would be worthwhile exploring ways to determine their values automatically per set of documents or per genre.

References

Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization (ISTS 1997), Madrid, pp. 10–17 (1997)

Bernard, J.R.L. (ed.): The Macquarie thesaurus. Macquarie Library, Sydney (1986)

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., Tan, Y.F.: The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics. In: Proceedings of Language Resources and Evaluation Conference (LREC 2008), Marrakesh, Morocco (May 2008)

Budanitsky, A., Hirst, G.: Evaluating WordNet-based measures of semantic distance. Computational Linguistics 32(1), 13–47 (2006)

Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In: Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference, pp. 26–33. Morgan Kaufmann Publishers Inc., San Francisco (2000)

Choi, F.Y.Y.: JTextTile: A free platform independent text segmentation algorithm. Software (1999), http://www.cs.man.ac.uk/~choif

Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication Series. MIT Press, Cambridge (1998)

Grosz, B.J., Sidner, C.L.: Attention, Intentions, and the Structure of Discourse. Computational Linguistics 12(3), 175–204 (1986)

Halliday, M.A.K., Hasan, R.: Cohesion in English. Longman, London (1976)

Hearst, M.A.: Multi-paragraph segmentation of expository text. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, USA. Association for Computational Linguistics (June 1994)

Hearst, M.A.: TextTiling: Segmenting text into multi-paragraph subtopic passages. Computational Linguistics 23(1), 33–64 (1997)

Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (ed.) WordNet: An electronic lexical database, pp. 305–332. The MIT Press, Cambridge (1998)

Hollingsworth, W.A.: Using Lexical Chains to Characterise Scientific Text. PhD thesis, Clare Hall College, University of Cambridge (2008)

Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference on Research on Computational Linguistics (ROCLING X), Taiwan (1997)

Kan, M.-Y., Klavans, J.L., McKeown, K.R.: Linear segmentation and segment significance. In: Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6), Montreal, Quebec, Canada, August 1998, pp. 197–205 (1998)

Kiss, G.R., Armstrong, C., Milroy, R., Piper, J.: An associative thesaurus of English and its computer analysis. In: Aitken, A.J., Bailey, R.W., Hamilton-Smith, N. (eds.) The Computer and Literary Studies. University Press, Edinburgh (1973)

Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: SIGDOC 1986: Proceedings of the 5th annual international conference on Systems documentation, pp. 24–26. ACM, New York (1986)

Lin, D.: Automatic Retrieval and Clustering of Similar Words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Montreal, Quebec, Canada, August 1998, vol. 2, pp. 768–774. Association for Computational Linguistics (1998a)

Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML 1998: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann Publishers Inc., San Francisco (1998b)

Mohammad, S., Hirst, G.: Distributional measures as proxies for semantic relatedness (2005), http://ftp.cs.toronto.edu/pub/gh/Mohammad+Hirst-2005.pdf

Mohammad, S., Hirst, G.: Distributional measures of concept-distance: A task-oriented evaluation. In: Proceedings, 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), Sydney, Australia (July 2006)

Morris, J., Hirst, G.: Lexical cohesion, the thesaurus, and the structure of text. Computational Linguistics 17(1), 21–48 (1991)

Okumura, M., Honda, T.: Word sense disambiguation and text segmentation based on lexical cohesion. In: COLING 1994: The 15th International Conference on Computational linguistics, Kyoto, Japan, vol. 2, pp. 755–761 (1994)

Passonneau, R.J., Litman, D.J.: Intention-based Segmentation: Human Reliability and Correlation with Linguistic Cues. In: Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio, USA, June 1993, pp. 148–155. Association for Computational Linguistics (1993)

Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet:: Similarity – Measuring the Relatedness of Concepts. In: Marcu, D., Dumais, S., Roukos, S. (eds.) HLT-NAACL 2004: Demonstration Papers, Boston, Massachusetts, USA, May 2004, pp. 38–41. Association for Computational Linguistics (2004)

Pevzner, L., Hearst, M.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics 28, 1–19 (2002)

Reynar, J.C.: Topic segmentation: Algorithms and applications. PhD thesis, Computer and Information Science, University of Pennsylvania (1998)

Stokes, N., Carthy, J., Smeaton, A.F.: SeLeCT: a lexical cohesion based news story segmentation system. AI Communications 17(1), 3–12 (2004)

Weeds, J.E.: Measures and applications of lexical distributional similarity. PhD thesis, University of Sussex (September 2003)

Yang, D., Powers, D.M.W.: Word Sense Disambiguation Using Lexical Cohesion in the Context. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, Australia, July 2006, pp. 929–936. Association for Computational Linguistics (2006)