The Design of a Configurable Text Summarization System

Ken BARKER   Yllias CHALI   Terry COPECK
Stan MATWIN   Stan SZPAKOWICZ

School of Information Technology and Engineering
University of Ottawa

TR-98-04

July 22, 1998

Abstract

This report presents the design of a flexible summarization system consisting of several independent linguistic processing tools that can be rapidly configured and extensively parameterized. Summarization will begin with segmentation of the text at places where there is a probable topic change. Next, segments will be classified and segments which strongly evidence of the topic suggested by a user's query will be identified. Finally, summary sentences will be extracted from the most relevant segments rather than the whole text.

1 Introduction

The Text Summarization (TS) part of the Intelligent Information Access (IIA) project (Lankester *et al.*, 1998) aims at generating summaries of text along the following dimensions:

- query-based summarization rather than generic summarization,
- indicative summaries rather than informative summaries,
- single rather than multiple documents, and
- extraction rather than abstraction.

We are designing a flexible summarization system consisting of several independent linguistic processing tools that can be rapidly configured and extensively parameterized. The top-level organization of the system is shown in Figure 1. It comprises the following generic modules: text segmenter, segment classifier and sentence extractor.

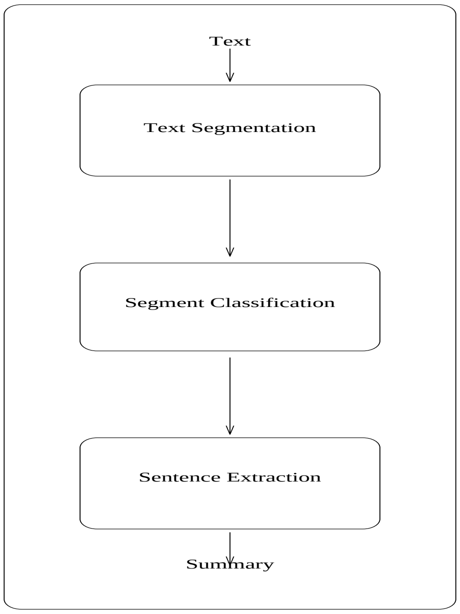

Figure 1: Text summarization system overview

Each module performs one major step in text summarization. A step may employ various complementary methods, and we expect to produce at least two different realizations of every module. For example, there will be two different text segmentation system at our disposal, two or more methods of extracting key phrases or salient technical terms, and so on. We will experiment extensively with the various possible three-step specific incarnations of the generic system. The overall organization of the experiments is shown in Figure 2. We will work with a fixed body of texts, and will use human judges to evaluate the effect of a parameter setting on the quality of the summaries. A text may come from a particular genre; we hope to discover useful correlations between parameterizations and genres. In a single experiment, we will select a configuration of the system, pick a number of texts, summarize them and judge the summaries. The relationship between the configuration choices and parameters and the quality of the summaries will be noted. We are designing a system in order to experiment with configurations and summary quality according to text characteristics, perhaps including genre (Delannoy *et al.*, 1998).

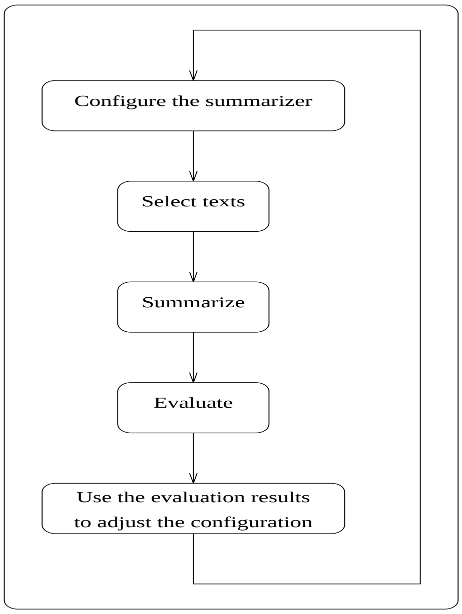

Figure 2: Configuration of the experiments with the text summarization system

Each configuration can be characterized by paths, in a directed acyclic graph (DAG), which describe choices of methods (see Figure 3). Each configuration is one path in the DAG.

2 Text segmentation

The purpose of this step is to subdivide a text into multi-paragraph units which are distinguished by subtopic shifts. Two public domain segmenters will be

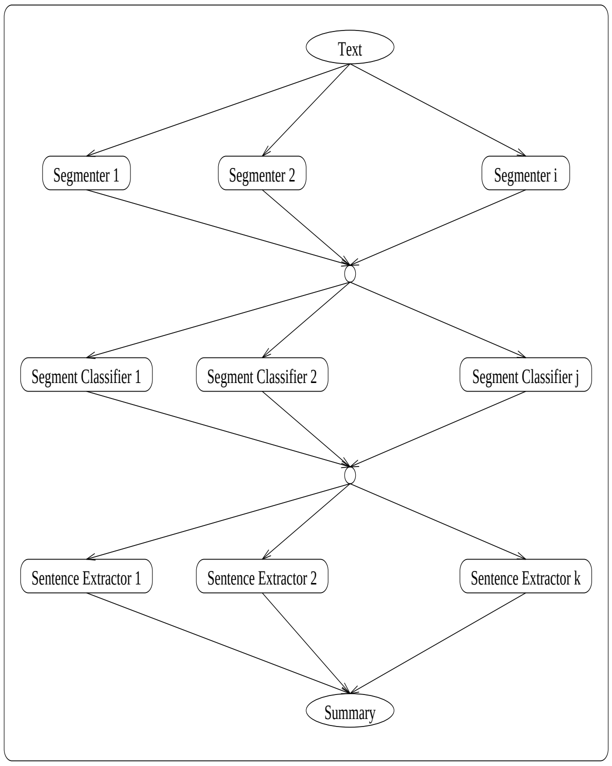

Figure 3: The different choices of configuration

considered for this step: the TextTiling system (Hearst, 1997) and a segmenter from the NLP group at Columbia University.

TextTiling employs a technique for subdividing texts into multi-paragraph units that represent passages or subtopics. The discourse cues for identifying major subtopic shifts are patterns of lexical co-occurrence and distribution. The algorithm casts texts into a linear sequence of segments. It compares adjacent pairs of text blocks for overall lexical similarity (based on word repetition) and detects gaps between adjacent text blocks. The blocks act as moving windows over the text. Several sentences can be contained within a block, but the blocks shift by only one sentence at the time.

Columbia's segmenter tool[1] detects topic changes by looking at noun phrases and their distribution. It uses the notion of word repetition chains to calculate boundaries and zero-sum weighting to arrive at the final calculation of borders. Word repetition chains are simply the stringing together of occurrences of the same stem word (after normalization of plurals to singulars, inflected verb and adjective forms to root form) into a "chain" based on their closeness within the text. The distribution of each phrase (common noun, pronoun or proper noun

[1] See http://www.cs.columbia.edu/˜min/research/segmenter/

phrase) affects the possible segment breaks.

Multi-paragraph subtopic segmentation should be useful for many text analysis tasks, including information retrieval and summarization. In our case, text segmentation is interesting for the following purposes:

- Segmentation is intended to identify the boundaries between paragraphs in a text where the text changes topic. Thus, a text can comprise merely a single segment, or perhaps several different segments, when it touches on several different topics.
- It helps the process of answering user queries in the sense that only segments that are relevant to the query terms are chosen for summarization.
- The resulting summaries can be more focused because the system limits itself to summarizing text segment units which deal with one topic rather than the summarization of largest text units or the whole documents which can involve several topic shifts. Shifts of topic in a summary are confusing.
- The summary can be smoother if it comes from a segment rather than from a whole text, a higher cohesion can be expected if only some segments are considered.

3 Segment classification

In order to extract the relevant sentences from the text segment units, we have to characterize the salient content by supplying the text segment units with substantial refinements of this level of analysis in discourse structure based on subtopic shift.

Several purely statistical and heuristic methods of characterization have been grounded in the information retrieval technology. These shallow methods, which do not involve any understanding of texts, seem limited by their inability to account for a number of linguistic phenomena–synonymy, polysemy, anaphora, metaphors, metonymy and context sensitivity. Our aim is to build a system which performs significantly better than those based only on statistical and heuristic methods. To this end we suggest:

- a method based on lexical cohesion
- a method based on noun phrase distribution

3.1 Lexical cohesion

The notion of cohesion, introduced in (Halliday and Hasan, 1976), is a device for "sticking together" different parts of the text to function as a whole. This "sticking together" is achieved through the use of reference, substitution, ellipsis, conjunction and semantically related words. Among these different means, lexical cohesion arises from semantic relationships between words. (Halliday and

Hasan, 1976) identify two categories of lexical cohesion: (a) reiteration, and (b) collocation. Reiteration can be achieved by the use of repetition, synonyms or near-synonyms, superordinates and general words which express generic semantic concepts. Collocation identifies relations between words that tend to co-occur in the same lexical context. Collocation relations are more problematic than reiteration because the former are not classifiable in any systematic fashion, but are instead rooted in the similarity that exists in situations or contexts in the world. Both of these categories are identifiable at the surface of the text. Lexical cohesion occurs both within the same sentence and across sentence boundaries. It is largely independent of the grammatical structure of the sentence. The effect of lexical cohesion is not limited to a pair of words, but it is very common for long cohesive chains among sequences of related words, called lexical chains (Morris and Hirst, 1991).

3.1.1 Lexical chains

The first computational model for lexical chains was presented in (Morris and Hirst, 1991). They define lexical cohesion relations in terms of categories, index entries and pointers to other categories in Roget's Thesaurus. Chains are created by taking a new text word and finding a chain related to it according to relatedness criteria. They analyze factors contributing to chain strength such as reiteration, density and length. They also introduce the notion of "activated chain" and "chain return" in order to take into account the distance between occurrences of related words. The authors built their chains by hand, and automation was not possible for lack of a machine-readable copy of the thesaurus.

More recently, (Barzilay and Elhadad, 1997) implemented an algorithm for the calculation of lexical chains using WordNet lexical database to determine the relatedness of the words (Miller *et al.*, 1993). Senses in the WordNet database are represented relationally by synonym sets–"synsets"–which are the sets of all the words sharing a common sense. Synsets exist for English nouns, verbs and adjectives, each representing one underlying lexical concept. WordNet expresses two types of relationship: lexical relations such as synonymy and antonymy, and semantic relations such as hyponymy and hypernymy. (Barzilay and Elhadad, 1997) introduce a kind of word sense disambiguation by considering all the senses of polysemous words in the construction of lexical chains. Assuming the text is cohesive, once the words in a chain have been connected, the best interpretation of the lexical concepts is the interpretation with the most connections.

For our text summarization system, we are considering a technique for lexical cohesion that resembles Barzilay and Elhadad's implementation in the sense that it is based on WordNet and uses WordNet's relations for gathering words in chains.

3.1.2 Lexical chain boundaries

Structural theories of text are concerned with identifying units of text that are about the "same thing". When this happens, there is a strong tendency for semantically related words to be used within that unit. By definition, lexical chains are chains of semantically related words, therefore under the assumption that texts are cohesive it makes sense to use them as clues to the structure of the text.

The intuition here is that calculating lexical chain boundaries will refine the structure of text segment units and perhaps identify a topic.

Halliday and Hasan(1976) argue that lexical cohesion occurs not simply between pairs of words but over a succession of a number of nearby related words spanning a topical unit of the text. There is distance relation between each word in the chain, and the words co-occur within a given span (Morris and Hirst, 1991). Lexical chains do not stop at sentence boundaries, they can connect a pair of adjacent words or range over an entire text. Morris and Hirst(1991) state that lexical chains tend to delineate portions of text that have a strong unity of meaning, so they tend to indicate the structure of the text, especially its linguistic segmentation. When a lexical chain ends, there is a tendency for a linguistic segment to end, as lexical chains tend to indicate segment topics. If a new lexical chain begins, this is a clue that a new segment has begun. If an old chain is referred to again (i.e. a chain return), it is a strong indication that a previous segment is being returned to.

In this step we augment lexical chains with distance relation, i.e. entities which correspond to the positions of lexical chain terms in the original text. The extent of the lexical chain is delimited by the positions of its first and last terms. The purpose of processing here is to analyze and identify the correspondence between lexical chain boundaries and structural unit boundaries.

3.1.3 Example

Consider the following text:

(A) With its distant orbit - 50 percent farther from the sun than Earth - and slim atmospheric blanket, Mars experiences frigid weather conditions. Surface temperatures typically average about -60 degrees Celsius (-76 degrees Fahrenheit) at the equator and can dip to -123 degrees C near the poles. Only the midday sun at tropical latitudes is warm enough to thaw ice on occasion, but any liquid water formed in this way should evaporate almost instantly because of the low atmospheric pressure.

Its lexical chains are:

(1) [orbit, sun, Earth, atmospheric, Mars, equator, poles, tropical, latitudes, atmospheric]

(2) [frigid, weather, temperatures, degrees Celsius, degrees Fahrenheit, degrees C]

(3) [warm, thaw, ice, liquid, water, evaporate, atmospheric pressure]

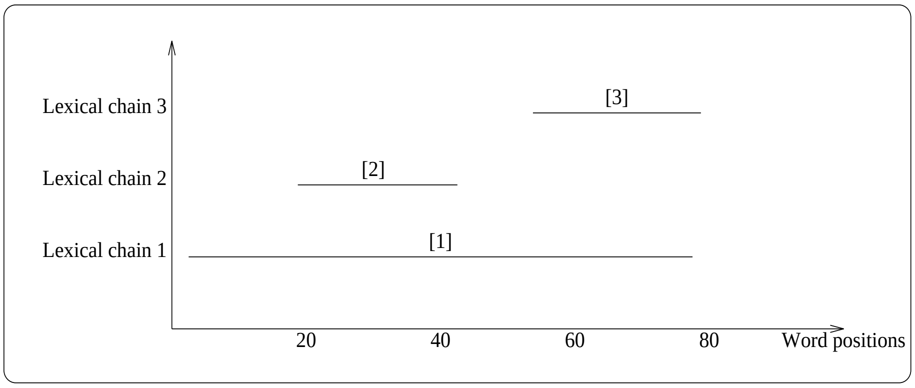

Figure 4: Distribution of the lexical chains of text (A)

So, if the extents of the lexical chains (1), (2) and (3) is represented respectively by the distance intervals [1], [2] and [3], as shown in Figure 4, we have the following relations[2]: $[2] \subset [1]$, $[3] \subset [1]$ and $[2] \prec [4]$, and the structural units corresponding to the above lexical chains can be described as follows:

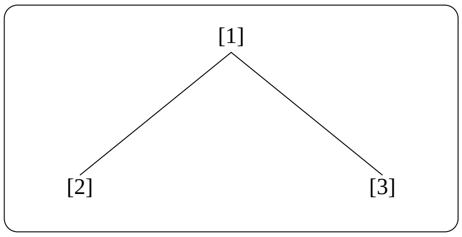

Figure 5: Structural units of text (A)

So, the lexical chain (1) covers the whole text segment (A), and within this segment two lexical chains (i.e. (2) and (3)) occur sequentially.

3.2 Noun phrase distribution

Noun phrase distribution is an alternative to the lexical chain method. It can be briefly described by the following steps:

- Extract the noun phrases from each text segment unit separately.
- Lookup and compute the distribution and the density of the noun phrases within the text segment unit. The purpose of this process is to detect the portions of the text segment that contain important information, and to characterize the dense portions.

[2] $\subset$ stands for subset and $\prec$ stands for precede.

- Experiment with several techniques and modes of ranking in order to select the dense portions of the text segment units. Several features contribute to selecting a sentence, in particular stochastic measurements of noun phrases, their location in the source text, the presence of cue or indicator phrases, or of title words.

This method is a preliminary design of a segment classifier and more work will be done on this later.

4 Sentence extraction

4.1 Preprocessing

Documents received by the text summarization subsystem are assumed to be in 7-bit ASCII with headings and paragraphs marked off by carriage returns. The Active Browser makes realistic assumptions about the format and encoding of documents it operates on (Lankester *et al.*, 1998), and sentence endings are not expected to be explicitly indicated in any way because end of sentence punctuation marks are ambiguous. An essential initial task is therefore to locate the boundaries of sentences; a second is to correctly distinguish the word tokens that make them up. These operations are done in the context of identifying the noun phrases and other significant text constituents and characteristics that serve as raw material for the summarization algorithm. They are performed using a version of the DIPETT parser (Delisle, 1994) tuned to the particular purposes and performance requirements of this project. Tuning involved giving priority to identifying noun phrases in particular and to doing so quickly. The two objectives are complementary.

Although simple shell scripts can do a job satisfactory for many purposes, robust and comprehensive tokenization and sentence boundary identification are not trivial tasks (Grefenstette and Tapanainen, 1994). DIPETT retains token type (word, number, punctuation etc) and capitalization (low, all-caps etc) information as it performs these operations, which include a limited amount of word translation to canonical forms (upper to lower case, contractions to full forms). These results and the noun phrases DIPETT extracts by a shallow parsing of the text are stored in a sentence database along with a variety of other information used by the summarization algorithm and the parent active browser. Items recorded include:

- the original sentence string, for display purposes;
- the offset in bytes from the beginning of the file to the beginning of the sentence;
- lists of tokenized noun phrases and other significant words such as discourse markers together with their frequencies;
- data computed by the summarization algorithm, which might include phrase densities, lexical chain labels and segment labels;

- the sentence rank as a candidate for inclusion in a summary as determined by the summarization algorithm;

The sentence database is the primary repository of information about the text on a sentence by sentence basis.

4.2 Extraction

Before the step of sentence extraction, it may be useful to provide various text extracts on demand by performing a text traversal which implies a type of reading where an area of interest is specified, and the best text segments representing that area are chosen in response.

Sentence extraction is based on text segment traversal. In the case of the noun phrase distribution method, text segment traversal is managed by scoring sentences, with the highest scorers included in the summary. In the case of the lexical cohesion method, text segment traversal is managed by paths in the structural units of text. It is necessary to distinguish paths that operate globally on a complete text segment unit from those restricted to some structural units within it.

The method of generating summaries by extracting sentences from the source text has several drawbacks, including the following main problems:

- Extraction of long sentences which include many constituents which would not have been selected on their own merit. An alternative can be considered: it involves some parsing of the sentences. A shallow syntactic parser can help on the one hand by extracting only the central constituents of the source text, and on the other hand by simplifing long, complex sentences (Elhadad *et al.*, 1997). The DIPETT parser can be adapted to meet the needs and the efficiency objectives of the summarizer.
- Extraction of sentences containing anaphoric links to the rest of the text. Heuristics have been proposed in the literature to address this problem, for example, by including with the extracted sentence the one that immediately preceding it. The best solution would be to replace anaphora with the referents using a shallow algorithm of anaphora resolution (Boguraev and Kennedy, 1996).

5 Summary evaluation

Evaluation is a key part of any research and development effort. We will employ two types: manual evaluation meant to help fine-tune configurations (see section 1), and task-oriented for satisfying user needs. Several tasks have been defined in Text Summarization (SUMMAC) competition which address different types of summaries (i.e. generic *vs* query-based, indicative *vs* informative)[3]. Two categories of evaluation method have been identified:

[3] See http://www.tipster.org

- intrinsic methods, i.e. generic summary based evaluation, and
- extrinsic methods, i.e. task or user- or query- or goal-directed summary based evaluation.

Given our system dimensions (see section 1), our system seems likely to be evaluated by methods in the second category. In this category, two tasks have been identified: categorization and ad hoc. In categorization, the goal will be to decide quickly whether or not a document contains information about any of a limited number of topic areas. This kind of task is not suitable to our system because it requires a generic summary. In the ad hoc task, a summarization system is considered as a back end to an information retrieval engine. An indicative summary may be used as an initial indicator of relevance prior to reviewing the full text of a document, and possibly eliminating the need to view that full text. Accordingly, our evaluation process must simulate the ad hoc scenario and we may have good summaries if we work for a good while with a fixed set of documents, as is our plan.

Evaluating systems that generate a summary tailored to a user query and intended to help him judge the relevance of a retrieved document requires measures for the following parameters (Firmin Hand, 1997):

- Accuracy or relevance decisions that are given by an assessor. Single or multiple assessors can be considered.
- Time required to make a relevance decision using a summary. This can be compared to the time required to make the same decision using the full text.
- Summary length as an optimal cutoff, e.g. 10 or 20% of full document length. However this characteristic seems invalidated by Jin *et al.*'s experiments (1998) that show (a) that there is no correlation between length and improvement in task (i.e. some systems have much higher precision for longer summaries, while human-generated summaries have a much higher precision for shorter summaries), (b) the time is not proportional to the length of the summary (e.g. a 10% summary does not require only 10% of the time needed to process the full text, and for some systems the time spent can decrease as the length of summaries increase). Rather they suggest eliminating the cutoff length requirement and allowing systems to set their own lengths, i.e. summarization systems that help with the task more and in less time are most suitable, no matter how long the summaries are.
- User preference evaluators will be asked whether they prefer the full text or the summary as a basis for decision-making, and will be encouraged to provide feedback as to why the summary was or was not acceptable for a given task. But Jin *et al.*'s experiments (1998) show that people's confidence in making decision can also increase or decrease with length of summaries.

Jin *et al.*(1998) introduces other measures such as: (a) query type, i.e. "easy" (all systems do equally well on this query) and "hard". However, recognition of the type of the query seems a difficult problem, (b) text selection, i.e. the relevance of a text to a query is sometimes only apparent by examining keywords extracted from it, (c) title, i.e. titles can themselves be very good indicative summaries, and judges may be able to decide the relevance of documents simply by reading titles alone.

Reviewing these considerations, we propose to use time and accuracy alone as performance measures.

6 Conclusion

This report has sketched the design of the top level of the text summarization system. It is based on pluggable modules that could easily be exchanged for versions that are improved in response to experiments.

The text summarization system is a part of the larger system of IIA, where it serves as a back end to the information retrieval engine: the user can quickly and accurately decide the relevancy of retrieved documents returned as a result of a query. The whole system provides the user with other tools that can help him to access the information such as keywords and titles (Lankester *et al.*, 1998).

The next step in our project is to implement a first version of the text summarization system, and to evaluate its performance and its efficiency according to summary evaluation criteria and methods to identify the best configurations.

References

(Barzilay and Elhadad, 1997) Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, 1997.

(Boguraev and Kennedy, 1996) Branimir Boguraev and Christopher Kennedy. Anaphora in a wider context: Tracking discourse referents. In *Proceedings of the 12th European Conference on Artificial Intelligence*, pages 582–586, 1996.

(Delannoy *et al.*, 1998) Jean François Delannoy, Ken Barker, Terry Copeck, Martin Laplante, Stan Matwin, and Stan Szpakowicz. Flexible summarization. In *AAAI 98 Spring Symposium on Intelligent Text Summarization*, pages 93–100, 1998.

(Delisle, 1994) Sylvain Delisle. *Text Processing without A-Priori Domain Knowledge: Semi-Automatic Linguistic Analysis for Incremental Knowledge Acquisition*. PhD thesis, Department of Computer Science, University of Ottawa, 1994.

(Elhadad *et al.*, 1997) Michael Elhadad, Kathleen McKeown, and Jacques Robin. Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–239, 1997.

(Firmin Hand, 1997) Therese Firmin Hand. A proposal for task-based evaluation of text summarization systems. In *ACL/EACL Workshop on Intelligent Scalable Text Summarization*, pages 31–38, 1997.

(Grefenstette and Tapanainen, 1994) Gregory Grefenstette and Pasi Tapanainen. What is a word, what is a sentence? problems of tokenization. Technical report, Rank Xerox Research Centre, Grenoble Laboratory, 1994.

(Halliday and Hasan, 1976) Michael Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman Group Ltd, 1976.

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

(Jin *et al.*, 1998) Hongyan Jin, Regina Brazilay, Kathleen McKeown, and Michael Elhadad. Summarization evaluation methods: Experiments and analysis. In *AAAI 98 Spring Symposium on Intelligent Text Summarization*, pages 60–68, 1998.

(Lankester *et al.*, 1998) Chris Lankester, Berry Debruijn, and Robert Holte. Prototype system for intelligent information access: Specification and implementation. Technical Report TR-98-05, School of Information Technology and Engineering, University of Ottawa, 1998.

(Miller *et al.*, 1993) George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Five papers on wordnet. CSL Report 43, Cognitive Science Laboratory, Princeton University, 1993.

(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.