Discourse Constraints for Document Compression

James Clarke*
University of Illinois at
Urbana-Champaign

Mirella Lapata**
University of Edinburgh

Sentence compression holds promise for many applications ranging from summarization to subtitle generation. The task is typically performed on isolated sentences without taking the surrounding context into account, even though most applications would operate over entire documents. In this article we present a discourse-informed model which is capable of producing document compressions that are coherent and informative. Our model is inspired by theories of local coherence and formulated within the framework of integer linear programming. Experimental results show significant improvements over a state-of-the-art discourse agnostic approach.

1. Introduction

Recent years have witnessed increasing interest in sentence compression. The task encompasses automatic methods for shortening sentences with minimal information loss while preserving their grammaticality. The popularity of sentence compression is largely due to its relevance for applications. Summarization is a case in point here. Most summarizers to date aim to produce informative summaries at a given compression rate. If we can have a compression component that reduces sentences to a minimal length and still retains the most important content, then we should be able to pack more information content into a fixed size summary. In other words, sentence compression would allow summarizers to increase the overall amount of information extracted without increasing the summary length (Lin 2003; Zajic et al. 2007). It could also be used as a post-processing step in order to render summaries more coherent and less repetitive (Mani, Gates, and Bloedorn 1999).

Beyond summarization, a sentence compression module could be used to display text on small screen devices such as PDAs (Corston-Oliver 2001) or as a reading aid for the blind (Grefenstette 1998). Sentence compression could also benefit information retrieval by eliminating extraneous information from the documents indexed by the

* Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, IL 61801, USA. E-mail: clarkeje@illinois.edu.
** School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, UK. E-mail: mlap@inf.ed.ac.uk.

Submission received: 10 September 2008; revised submission received: 27 October 2009; accepted for publication: 6 March 2010.

retrieval engine. This way it would be possible to store less information in the index without dramatically affecting retrieval performance (Olivers and Dolan 1999).

In theory, sentence compression may involve several rewrite operations such as deletion, substitution, insertion, and word reordering. In practice, however, the task is commonly defined as a word deletion problem: Given an input sentence of words $\mathbf{x} = x_1, x_2, \ldots, x_n$, the aim is to produce a compression by removing any subset of these words (Knight and Marcu 2002). Many sentence compression models aim to learn deletion rules from a parsed parallel corpus of **source** sentences and their **target** compressions (Knight and Marcu 2002; Turner and Charniak 2005; Galley and McKeown 2007; Cohn and Lapata 2009). For example, Knight and Marcu (2002) learn a synchronous context-free grammar (Aho and Ullman 1969) from such a corpus. The grammar rules have weights (essentially probabilities estimated using maximum likelihood) and are used to find the best compression from the set of all possible compressions for a given sentence. Other approaches exploit syntactic information without making explicit use of a parallel grammar—for example, by learning which words or constituents to delete from a parse tree (Riezler et al. 2003; Nguyen et al. 2004; McDonald 2006; Clarke and Lapata 2008).

Despite differences in formulation and training requirements (some approaches require a parallel corpus, whereas others do not), existing models are similar in that they compress sentences in isolation without taking their surrounding context into account. This is in marked contrast with common practice in summarization. Professional abstractors often rely on contextual cues while creating summaries (Endres-Niggemeyer 1998). This is true of automatic summarization systems too, which consider the position of a sentence in a document and how it relates to its surrounding sentences (Kupiec, Pedersen, and Chen 1995; Barzilay and Elhadad 1997; Marcu 2000; Teufel and Moens 2002). Determining which information is important in a sentence is not merely a function of its syntactic position (e.g., deleting the verb or the subject of a sentence is less likely). A variety of contextual factors can play a role, such as the discourse topic, whether the sentence introduces new entities or events that have not been mentioned before, or the reader's background knowledge.

A sentence-centric view of compression is also at odds with most relevant applications which aim to create a shorter document rather than a single sentence. The resulting document must not only be grammatical but also coherent if it is to function as a replacement for the original. However, this cannot be guaranteed without knowledge of how the discourse progresses from sentence to sentence. To give a simple example, a contextually aware compression system could drop a word or phrase from the current sentence, simply because it is not mentioned anywhere else in the document and is therefore deemed unimportant. Or it could decide to retain it for the sake of topic continuity.

In this article we are interested in creating a compression model that is appropriate for both documents and sentences. Luckily, a variety of discourse theories have been developed over the years (e.g., Mann and Thompson, 1988; Grosz, Weinstein, and Joshi 1995; Halliday and Hasan 1976) and have found application in summarization (Barzilay and Elhadad 1997; Marcu 2000; Teufel and Moens 2002) and other text generation applications (Scott and de Souza 1990; Kibble and Power 2004). In creating a context-sensitive compression model we are faced with three important questions: (1) Which type of discourse information is useful for compression? (2) Is it amenable to automatic processing (there is little hope for interfacing our compression model with applications if discourse-level cues cannot be identified robustly)? and (3) How are sentence- and document-based information best integrated in a unified modeling framework?

In building our compression model we borrow insights from two popular models of discourse, Centering Theory (Grosz, Weinstein, and Joshi 1995) and lexical chains (Morris and Hirst 1991). Both approaches capture **local coherence**—the way adjacent sentences bind together to form a larger discourse. They also both share the view that discourse coherence revolves around discourse entities and the way they are introduced and discussed. We first automatically augment our documents with annotations pertaining to centering and lexical chains, which we subsequently use to inform our compression model. The latter is an extension of the integer linear programming formulation proposed by Clarke and Lapata (2008). In a nutshell, sentence compression is modeled as an optimization problem. Given a long sentence, a compression is formed by retaining the words that maximize a scoring function coupled with a small number of constraints ensuring that the resulting output is grammatical. The constraints are encoded as linear inequalities whose solution is found using integer linear programming (ILP; Winston and Venkataramanan 2003; Vanderbei 2001). Discourse-level information can be straightforwardly incorporated by slightly changing the compression objective—we now wish to compress entire documents rather than isolated sentences—and augmenting the constraint set with discourse-specific constraints. We use our model to compress whole documents (rather than sentences sequentially) and evaluate whether the resulting text is understandable and informative using a question-answering task. We show that our method yields significant improvements over discourse agnostic state-of-the-art compression models (McDonald 2006; Clarke and Lapata 2008).

The remainder of this article is organized as follows. Section 2 provides an overview of related work. In Section 3 we present the ILP framework and compression model we employ in our experiments. We introduce our discourse-related extensions in Sections 4 and 5. Section 6 discusses our experimental set-up and evaluation methodology. Our results are presented in Section 7. Discussion of future work concludes the paper.

2. Related Work

Sentence compression has been extensively studied across different modeling paradigms and has received both generative and discriminative formulations. Most generative approaches (Knight and Marcu 2002; Turner and Charniak 2005; Galley and McKeown 2007) are instantiations of the noisy-channel model, whereas discriminative formulations include decision-tree learning (Knight and Marcu 2002), maximum entropy (Riezler et al. 2003), support vector machines (Nguyen et al. 2004), and large-margin learning (McDonald 2006; Cohn and Lapata 2009). These models are trained on a parallel corpus and learn either which constituents to delete or which words to place adjacently in the compression output. Relatively few approaches dispense with the parallel corpus and generate compressions in an unsupervised manner using either a scoring function (Hori and Furui 2004; Clarke and Lapata 2008) or compression rules that are approximated from a non-parallel corpus such as the Penn Treebank (Turner and Charniak 2005).

The majority of sentence compression approaches only look at sentences in isolation without taking into account any discourse information. However, there are two notable exceptions. Jing (2000) uses information from the local context as evidence for and against the removal of phrases during sentence compression. The idea here is that words or phrases which have more links to the surrounding context are more indicative of its topic, and thus should not be dropped. The topic is not explicitly identified; instead the importance of each phrase is determined by the number of lexical links within the local context. A link is created between two words if they are repetitions,

morphologically related, or associated in WordNet (Fellbaum 1998) through a lexical relation (e.g., hyponymy, synonymy). Links have weights—for example, repetition is considered more important than hypernymy. Each word is assigned a context weight based on the number of links to the local context and the importance of each relation type. Phrases are scored by the sum of their children's context scores. The decision to drop a phrase is influenced by several factors, besides the local context, such as the phrase's grammatical role and previous evidence from a parallel corpus.

Daumé III and Marcu (2002) generalize sentence compression to document compression. Given a document $D = w_1, w_2, \ldots, w_n$ the goal is to produce a **summary**, S, by dropping any subset of words from D. Their system uses the discourse structure of a document and the syntactic structure of each of its sentences in order to decide which words to drop. Specifically, they extend Knight and Marcu's (2002) noisy-channel model so that it can be applied to entire documents. In its simpler sentence compression instantiation, the noisy-channel model has two components, a language model and a channel model, both of which act on probabilistic context-free grammar (PCFG) representations. Daumé III and Marcu define a noisy-channel model over syntax and discourse trees. Following Rhetorical Structure Theory (RST; Mann and Thompson 1988), they represent documents by trees whose leaves correspond to elementary discourse units (*edus*) and whose nodes specify how these and larger units (e.g., multi-sentence segments) are linked to each other by rhetorical relations (e.g., *Contrast*, *Elaboration*). Discourse units are further characterized in terms of their text importance: **nuclei** denote central segments, whereas **satellites** denote peripheral ones. Their model therefore learns not only which syntactic constituents to drop but also which discourse units are unimportant.

While Daumé III and Marcu (2002) present a hybrid summarizer that can simultaneously delete words and sentences from a document, the majority of summarization systems to date simply select and present to the user the most important sentences in a text (see Mani [2001] for a comprehensive overview of the methods used to achieve this). Discourse-level information plays a prominent role here as the overall document organization can indicate whether a sentence should be included in the summary. A variety of approaches have focused on cohesion (Halliday and Hasan 1976) and the way it is expressed in discourse. The term broadly describes a variety of linguistic devices responsible for making the elements of a text appear unified or connected. Examples include word repetition, anaphora, ellipsis, and the use of synonyms or superordinates. The underlying assumption is that sentences connected to many other sentences are likely to carry salient information and should therefore be included in the summary (Sjorochod'ko 1972). In exploiting cohesion for summarization, it is necessary to somehow represent cohesive ties. For instance, Boguraev and Kennedy (1997) represent cohesion in terms of anaphoric relations, whereas Barzilay and Elhadad (1997) operationalize cohesion via **lexical chains**—sequences of related words spanning a topical unit (Morris and Hirst 1991). Besides repetition, they also examine semantic relations based on synonymy, antonymy, hypernymy, and holonymy (we discuss their approach in more detail in Section 4.1).

Other approaches characterize the document in terms of discourse structure and rhetorical relations. Documents are commonly represented as trees (Mann and Thompson 1988; Corston-Oliver 1998; Ono, Sumita, and Miike 1994; Carlson et al. 2001) and the position of a sentence in a tree is indicative of its importance. To give an example, Marcu (2000) proposes a summarization algorithm based on RST. Assuming that nuclei are more salient than satellites, the importance of sentential or clausal units can be determined based on tree depth. Alternatively, discourse structure can be represented as a graph (Wolf and Gibson 2004) and sentence importance is determined in

graph-theoretic terms, by using graph connectivity measures such as in-degree or PageRank (Brin and Page 1998). Although a great deal of research in summarization has focused on *global* properties of discourse structure, there is evidence that *local* coherence may also be useful without the added complexity of computing discourse representations. (Unfortunately, discourse parsers have yet to achieve levels of performance comparable to syntactic parsers.) Teufel and Moens (2002) identify discourse relations on a sentence-by-sentence basis without presupposing an explicit discourse structure. Inspired by Centering Theory (Grosz, Weinstein, and Joshi 1995)—a theory of local discourse structure that models the interaction of referential continuity and salience of discourse entities—Orăsan (2003) proposes a summarization algorithm that extracts sentences with at least one entity in common. The idea here is that summaries containing sentences referring to the same entity will be more coherent. Other work has relied on centering not so much to create summaries but to assess whether they are readable (Barzilay and Lapata 2008).

Our approach differs from previous sentence compression approaches in three key respects. First, we present a compression model that is contextually aware; decisions on whether to remove or retain a word (or phrase) are informed by its discourse properties (e.g., whether it introduces a new topic, or whether it is semantically related to the previous sentence). Unlike Jing (2000) we explicitly identify topically important words and assume specific representations of discourse structure. Secondly, in contrast to Daumé III and Marcu (2002) and other summarization work, we adopt a less global and more shallow representation of discourse based on Centering Theory and lexical chains. One of our aims is to exploit discourse features that can be computed efficiently and relatively cheaply. Thirdly, our compression model can be applied to isolated sentences as well as to entire documents. We claim the latter is more in the spirit of real-world applications where the goal is to generate a condensed and coherent text. Unlike Daumé III and Marcu (2002) our model can delete words but not sentences, although it could be used to compress documents of any type, even summaries.

3. The Compression Model

Our model is an extension of the approach put forward in Clarke and Lapata (2008) where they formulate sentence compression as an optimization problem. Given a long sentence, a compression is created by retaining the words that maximize a scoring function. The latter is essentially a language model coupled with a few constraints ensuring that the resulting output is grammatical. The language model and the constraints are encoded as linear inequalities whose solution is found using ILP.[1]

Their model is a good point of departure for studying document-based compression. As it does not require a parallel corpus, it can be ported across domains and text genres, while delivering state-of-the-art results (see Clarke and Lapata [2008] for details). Importantly, discourse-level information can be easily incorporated in two ways: Firstly, by applying the compression objective to entire documents rather than individual sentences; and secondly, by augmenting the constraint set with discourse-related information. This is not the case for other approaches (e.g., those based on the noisy channel model) where compression is modeled by grammar rules indicating which constituents to delete in a syntactic context. Moreover, ILP delivers a globally

1 It is outside the scope of this article to provide an introduction to ILP. We refer the interested reader to Winston and Venkataramanan (2003) and Vanderbei (2001) for comprehensive overviews.

optimal solution by searching over the entire compression space[2] without employing heuristics or approximations during decoding (see Turner and Charniak [2005] and McDonald [2006] for examples).

Besides sentence compression, the ILP modeling framework has been applied to a wide range of natural language processing tasks demonstrating improvements over more traditional methods. Examples include reluctant paraphrasing (Dras 1997), relation extraction (Roth and Yih 2004), semantic role labeling (Punyakanok et al. 2004), concept-to-text generation (Marciniak and Strube 2005; Barzilay and Lapata 2006), dependency parsing (Riedel and Clarke 2006; Martins, Smith, and Xing 2009), and coreference resolution (Denis and Baldridge 2007).

In the following we describe Clarke and Lapata's (2008) model in more detail. Sections 4–5 present our extensions and modifications.

3.1 Language Model

Let $\mathbf{x} = x_0, x_1, x_2, \ldots, x_n$ denote a source sentence for which we wish to generate a target compression. We use x_0 to denote the "start" token. We introduce a decision variable for each word in the source and constrain it to be binary; a value of 0 represents a word being dropped, whereas a value of 1 includes the word in the target compression. Let:

$$\delta_i = \begin{cases} 1 \text{ if } x_i \text{ is in the compression} \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in [1 \ldots n]$$

A trigram language model forms the backbone of the compression model. The language model is formulated as an integer linear program with the introduction of extra decision variables indicating which **word sequences** should be retained or dropped from the compression. Let:

$$\alpha_i = \begin{cases} 1 \text{ if } x_i \text{ starts the compression} \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in [1 \ldots n]$$

$$\beta_{ij} = \begin{cases} 1 \text{ if sequence } x_i, x_j \text{ ends} \\ \quad \text{the compression} & \forall i \in [0 \ldots n-1] \\ 0 \text{ otherwise} & \forall j \in [i+1 \ldots n] \end{cases}$$

$$\gamma_{ijk} = \begin{cases} 1 \text{ if sequence } x_i, x_j, x_k & \forall i \in [0 \ldots n-2] \\ \quad \text{is in the compression} & \forall j \in [i+1 \ldots n-1] \\ 0 \text{ otherwise} & \forall k \in [j+1 \ldots n] \end{cases}$$

The objective function is expressed in Equation (1). It is the sum of all possible trigrams multiplied by the appropriate decision variable where n is the length of the sentence (note all probabilities throughout this paper are log-transformed). The objective function also includes a significance score $I(x_i)$ for each word x_i multiplied by the decision

2 For a sentence of length n, there are 2^n compressions.

variable for that word (see the first summation term in Equation (1)). This score highlights important content words in a sentence and is defined in Section 3.2.

$$\begin{aligned}\max z = & \sum_{i=1}^{n} \delta_i \cdot \lambda I(x_i) + \sum_{i=1}^{n} \alpha_i \cdot P(x_i|\text{start}) \\ & + \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \gamma_{ijk} \cdot P(x_k|x_i, x_j) \\ & + \sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \beta_{ij} \cdot P(\text{end}|x_i, x_j) \\ & - \zeta_{min} \cdot \mu - \zeta_{max} \cdot \mu \end{aligned} \tag{1}$$

Note that we add a weighting factor, λ, to the objective, in order to counterbalance the importance of the language model and the significance score.

The final component of our objective function, $\zeta \cdot \mu$, relates to the compression rate. As we explain shortly (Equations (7) and (8)) the compressions our model generates are subject to a prespecified compression rate. For instance we may wish to create compressions at a minimum rate of 40% and maximum rate of 70%. The compression rate constraint can be violated with a penalty, μ, which applies to each word. ζ_{min} counts the number of words under the compression rate and ζ_{max} the number of words over the compression rate. Thus, the more the output violates the compression rate, the larger the penalty will be. In other words, the term $\zeta_{min} \cdot \mu - \zeta_{max} \cdot \mu$ acts as a soft constraint providing a means to guide the compression towards the desired rate. The violation penalty μ is tuned experimentally and may vary depending on the desired compression rate or application.

The objective function in Equation (1) allows any combination of trigrams to be selected. As a result, invalid trigram sequences (e.g., two or more trigrams containing the "end" token) could appear in the target compression. We avoid this situation by introducing **sequential constraints** (on the decision variables δ_i, γ_{ijk}, α_i, and β_{ij}) that restrict the set of allowable trigram combinations.

Constraint 1. Exactly one word can begin a sentence.

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{2}$$

Constraint 2. If a word is included in the sentence it must either start the sentence or be preceded by two other words or one other word and the "start" token x_0.

$$\delta_k - \alpha_k - \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} \gamma_{ijk} = 0 \tag{3}$$

$$\forall k : k \in [1 \ldots n]$$

Constraint 3. If a word is included in the sentence it must either be preceded by one word and followed by another or it must be preceded by one word and end the sentence.

$$\delta_j - \sum_{i=0}^{j-1} \sum_{k=j+1}^{n} \gamma_{ijk} - \sum_{i=0}^{j-1} \beta_{ij} = 0 \quad (4)$$
$$\forall j : j \in [1 \dots n]$$

Constraint 4. If a word is in the sentence it must be followed by two words or followed by one word and then the end of the sentence or it must be preceded by one word and end the sentence.

$$\delta_i - \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \gamma_{ijk} - \sum_{j=i+1}^{n} \beta_{ij} - \sum_{h=0}^{i-1} \beta_{hi} = 0 \quad (5)$$
$$\forall i : i \in [1 \dots n]$$

Constraint 5. Exactly one word pair can end the sentence.

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n} \beta_{ij} = 1 \quad (6)$$

Note that Equations (2)–(6) are merely well-formedness constraints and differ from the compression-specific constraints which we discuss subsequently. Any language model formulated as an ILP would require similar constraints.

Compression rate constraints. Depending on the application or the task at hand, we may require that the compressions fall within a specific compression rate. We assume here that our model is given a compression rate range, $c_{min}\% - c_{max}\%$, and create two constraints that penalize compressions which do not fall within this range:

$$\sum_{i=0}^{n} \delta_i + \zeta_{min} \geq c_{min} \cdot n \quad (7)$$

$$\sum_{i=0}^{n} \delta_i - \zeta_{max} \leq c_{max} \cdot n \quad (8)$$

Here, δ_i is still a decision variable for each word, n is the number of words in the sentence, ζ is the number of words over or under the compression rate, and c_{min} and c_{max} are the limits of the range.

3.2 Significance Score

The significance score is an attempt at capturing the gist of a sentence. The score has two components which correspond to document and sentence importance, respectively. Given a sentence and its syntactic parse, we define $I(x_i)$ as:

$$I(x_i) = f_i \log \frac{F_a}{F_i} \cdot \frac{l}{N} \quad (9)$$

where x_i is a topic word, f_i is x_i's document frequency, F_i its corpus frequency, and F_a the sum of all topic words in the corpus; l is the number of clause constituents above x_i, and N is the deepest level of clause embedding in the parse.

The first term in Equation (9) is similar to $tf * idf$; it highlights words that are important in the document and should therefore not be dropped. The score is not applied indiscriminately to all words in a sentence but solely to topic-related words, which are approximated by nouns and verbs. This is offset by the importance of these words in the specific sentence being compressed. Intuitively, in a sentence with multiply nested clauses, more deeply embedded clauses tend to carry more semantic content. This is illustrated in Figure 1, which depicts the clause embedding for the sentence *Mr Field has said he will resign if he is not reselected, a move which could divide the party nationally*. Here, the most important information is conveyed by clauses S_3 (*he will resign*) and S_4 (*if he is not reselected*), which are embedded. Accordingly, we should give more weight to words found in these clauses than in the main clause (S_1 in Figure 1). A simple way to enforce this is to give clauses weight proportional to the level of embedding (see the second term in Equation (9)). Therefore in Figure 1, the term $\frac{l}{N}$ is 1.0 (4/4) for clause S_4, 0.75 (3/4) for clause S_3, and so on. Individual words inherit their weight from their clauses. We obtain syntactic information in our experiments from RASP (Briscoe and Carroll 2002), a domain-independent, robust parsing system for English. However, any other parser with broadly similar output (e.g., Lin 2001) could also serve our purposes.

Note that the significance score in Equation (9) does not weight differentially the contribution of $tf * idf$ versus level of embedding. Although we found in our experiments that the latter term was as important as $tf * idf$ in producing meaningful compressions, there may be applications or data sets where the contribution of the two terms varies. This could be easily remedied by introducing a weighting factor.

3.3 Sentential Constraints

In its original formulation, the model also contains a small number of sentence-level constraints. Their aim is to preserve the meaning and structure of the original sentence as much as possible. The majority of constraints revolve around modification and

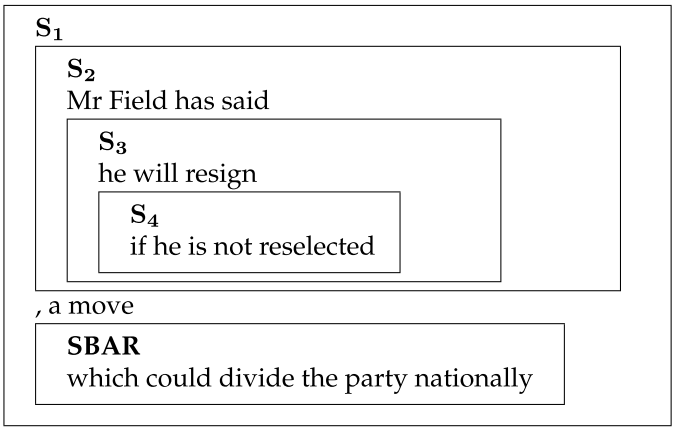

Figure 1
The clause embedding of the sentence *Mr Field has said he will resign if he is not reselected, a move which could divide the party nationally*; nested boxes correspond to nested clauses.

argument structure and are defined over parse trees or grammatical relations which as mentioned earlier we extract from RASP.

Modifier Constraints. Modifier constraints ensure that relationships between head words and their modifiers remain grammatical in the compression:

$$\delta_i - \delta_j \geq 0 \quad \forall i, j : x_j \in x_i\text{'s } \texttt{ncmods} \tag{10}$$

$$\delta_i - \delta_j \geq 0 \quad \forall i, j : x_j \in x_i\text{'s } \texttt{detmods} \tag{11}$$

Equation (10) guarantees that if we include a non-clausal modifier[3] (`ncmod`) in the compression (such as an adjective or a noun) then the head of the modifier must also be included; this is repeated for determiners (`detmod`) in Equation (11).

Other modifier constraints ensure the meaning of the source sentence is preserved in the compression. For example, Equation (12) enforces *not* in the compression when the head is included. A similar constraint is added for possessive modifiers (e.g., *his*, *our*), including genitives (e.g., *John's gift*), as shown in Equation (13).

$$\delta_i - \delta_j = 0 \quad \forall i, j : x_j \in x_i\text{'s } \texttt{ncmods} \wedge x_j = \text{not} \tag{12}$$

$$\delta_i - \delta_j = 0 \quad \forall i, j : x_j \in x_i\text{'s possessive } \texttt{mods} \tag{13}$$

Argument Structure Constraints. Argument structure constraints make sure that the resulting compression has a canonical argument structure. The first constraint (Equation (14)) ensures that if a verb is present in the compression then so are its arguments, and if any of the arguments are included in the compression then the verb must also be included.

$$\delta_i - \delta_j = 0 \quad \forall i, j : x_j \in \text{subject/object of verb } x_i \tag{14}$$

Another constraint forces the compression to contain at least one verb provided the source sentence contains one as well:

$$\sum_{i:x_i \in verbs} \delta_i \geq 1 \tag{15}$$

3 Clausal modifiers (`cmod`) are adjuncts modifying entire clauses. In the example *he ate the cake because he was hungry*, the *because*-clause is a modifier of the sentence *he ate the cake*.

Other constraints apply to prepositional phrases and subordinate clauses and force the introducing term (i.e., the preposition, or subordinator) to be included in the compression if any word from within the syntactic constituent is also included:

$$\delta_i - \delta_j \geq 0 \quad (16)$$
$$\forall i, j : x_j \in \texttt{PP/SUB} \wedge x_i \text{ starts } \texttt{PP/SUB}$$

By subordinator (SUB) we mean *wh*-words (e.g., *who, which, how, where*), the word *that*, and subordinating conjunctions (e.g., *after, although, because*). The reverse is also true—that is, if the introducing term is included, at least one other word from the syntactic constituent should also be included.

$$\sum_{i:x_i \in \texttt{PP/SUB}} \delta_i - \delta_j \geq 0 \quad (17)$$
$$\forall j : x_j \text{ starts } \texttt{PP/SUB}$$

All the constraints described thus far are mostly syntactic. They operate over parse trees or dependency graphs. In the following sections we present our discourse-specific constraints. But first we discuss how we represent and automatically detect discourse-related information.

4. Discourse Representation

Obtaining an appropriate representation of discourse is the first step toward creating a compression model that exploits document-level information. Our goal is to annotate documents automatically with discourse-level information which will subsequently be used to inform our compression procedure. As mentioned in Section 2 previous summarization work has mainly focused on cohesion (Sjorochod'ko 1972; Barzilay and Elhadad 1997) or global discourse structure (Marcu 2000; Daumé III and Marcu 2002). We also opt for a cohesion-based representation of discourse operationalized by lexical chains (Morris and Hirst 1991). Computing global discourse structure robustly and accurately is far from trivial. For example, Daumé III and Marcu (2002) employ an RST parser[4] but find that it produces noisy output for documents containing longer sentences. We therefore focus on the less ambitious task of characterizing local coherence—the way adjacent sentences bind together to form a larger discourse. Although it does not explicitly capture long distance relationships between sentences, local coherence is still an important prerequisite for maintaining global coherence. Specifically, we turn to Centering Theory (Grosz, Weinstein, and Joshi 1995) and adopt an entity-based representation of discourse.

In the following sections we briefly introduce lexical chains and centering and describe our algorithms for obtaining discourse annotations.

4 This is the decision-based parser described in Marcu (2000); it achieves an F1 of 38.2 for the identification of elementary discourse units, 50.0 for hierarchical spans, 39.9 for nuclearity, and 23.4 for relation assignment.

4.1 Lexical Chains

Lexical cohesion refers to the degree of semantic relatedness observed among lexical items in a document. The term was coined by Halliday and Hasan (1976), who observed that coherent documents tend to have more related terms or phrases than incoherent ones. A number of linguistic devices can be used to signal cohesion; these range from repetition, to synonymy, hyponymy, and meronymy. Lexical chains are a representation of lexical cohesion as sequences of semantically related words (Morris and Hirst 1991). There is a close relationship between discourse structure and cohesion. Related words tend to co-occur within the same discourse. Thus, cohesion is a surface indicator of discourse structure and can be identified through lexical chains.

Lexical chains provide a useful means for describing the topic flow in discourse. For example, a document containing the chain {*house*, *home*, *loft*, *house*} will probably describe a situation involving a house. Documents often have multiple topics (or themes) and consequently will contain many different lexical chains. Some of these topics will be peripheral and thus represented by short chains whereas main topics will correspond to dense longer chains. Words participating in the latter chains are important for our compression task—they reveal what the document is about—and in all likelihood should not be deleted.

Barzilay and Elhadad (1997) describe a technique for building lexical chains for extractive text summarization. In their approach chains of semantically related expressions are used to select sentences for inclusion in a summary. Their algorithm uses WordNet (Fellbaum 1998) to build chains of nouns (and noun compounds). Nouns are considered related if they are repetitions or linked in WordNet via synonymy, antonymy, hypernymy, and holonymy. Computing lexical chains would be relatively straightforward if each word was always represented by a single sense. However, due to the high level of polysemy inherent in WordNet, algorithms developed for computing lexical chains must adopt some strategy for disambiguating word senses. For example, Hirst and St-Onge (1998) greedily disambiguate a word as soon as it is encountered by selecting the sense most strongly related to existing chain members, whereas Barzilay and Elhadad (1997) consider all possible alternatives of word senses and then choose the best one among them.

Once created, lexical chains can serve to highlight which document sentences are more topical, and should therefore be included in a summary. Barzilay and Elhadad (1997) rank their chains heuristically by a score based on their length and homogeneity. They generate summaries by extracting sentences corresponding to **strong chains**, that is, chains whose score is two standard deviations above the average score. Analogously, we also wish to determine which lexical chains indicate the most prevalent discourse topics. Our assumption is that terms belonging to these chains are indicative of the document's main focus and should therefore be retained in the compressed output. Barzilay and Elhadad's (1997) scoring function aims to identify sentences (for inclusion in a summary) that have a high concentration of chain members. In contrast, we are interested in chains that span several sentences. We thus score chains according to the number of sentences their terms occur in. For example, the hypothetical chain {$house_3$, $home_3$, $loft_3$, $house_5$} (where $word_i$ denotes *word* occurring in sentence *i*) would be given a score of two as the terms occur only in two sentences. We assume that a chain signals a prevalent discourse topic if it occurs throughout more sentences than the average chain. The scoring algorithm is outlined more formally as:

1. Compute the lexical chains for the document.

2. $Score(Chain) = Sentences(Chain)$.
3. Discard chains for which $Score(Chain) < Average(Score)$.
4. Mark terms from the remaining chains as being the focus of the document.

We use the method of Galley and McKeown (2003) to compute lexical chains for each document.[5] It improves on Barzilay and Elhadad's (1997) original algorithm by providing better word sense disambiguation and linear runtime. The algorithm proceeds in three steps. Initially, a graph is built representing all possible interpretations of the document under consideration. The text is processed sequentially, comparing each word against all words previously read. If a relation exists between the senses of the current word and any possible sense of a previous word, a connection is formed between the appropriate words and senses. The strength of the connection is a function of the type of relationship and of the distance between the words in the text (in terms of words, sentences, and paragraphs). Words are represented as nodes in the graph and semantic relations as weighted edges. The relations considered by Galley and McKeown (2003) are all first-order WordNet relations, with the addition of **siblings**—two words are considered siblings if they are both hyponyms of the same hypernym. Next, all occurrences of a given word are collected together. For each sense of a target word, the strength of all connections involving that sense are summed, giving that sense a unified score. The sense with the highest unified score is chosen as the correct sense for the target word. Lastly, the lexical chains are constructed by collecting same sense words into the same chain.

Figure 2 illustrates the lexical chains created by our algorithm for three documents (taken from our test set). Chains are shown in oval boxes; members of the same chain have the same index. The algorithm identifies three chains in the first document: {*flow, rate*}, {*today, day, yesterday*}, and {*miles, ft*}. In the second document the chains are {*body*} and {*month, night*}, and in the third {*policeman, police*}, {*woman, woman, boyfriend, man*}. As can be seen, members of a chain represent a shared concept (e.g., "time", "linear unit", or "person"). In some cases important topics are missed. For instance, in the first document no chains were created with the words *lava* or *debris*. The second document is about *Mrs Allan* and contains many references to her. However, because *Mrs Allan* is not listed in WordNet it is not possible to create any chains for this word or any of its coreferents (e.g., *she, her*). A similar problem is observed in the third document where *Anderson* is not included in any chain even though he is one of the main protagonists throughout the text. We next turn to Centering Theory as a means of identifying which entities are prominent in a document.

4.2 Centering Theory

Centering Theory (Grosz, Weinstein, and Joshi 1995) is an entity-orientated theory of local coherence and salience. One of the main ideas underlying centering is that certain entities mentioned in an utterance are more central than others. This in turn imposes constraints on the use of referring expressions and in particular on the use of pronouns.

The theory begins by assuming that a discourse is broken into "utterances." These can be phrases, clauses, sentences, or even paragraphs. At any point in discourse, some entities are considered more salient than others, and are expected to exhibit

5 The software is available from `http://www1.cs.columbia.edu/nlp/tools.cgi`.

Bad (weather) dashed hopes of attempts to halt the (flow$_1$) during what was seen as a lull in the [lava's] momentum. Experts say that even if the eruption stopped (today$_2$), the pressure of [lava] piled up behind for six (miles$_3$) would bring [debris] cascading down on to the (town) anyway. Some estimate the volcano is pouring out one million tons of [debris] a (day$_2$), at a (rate$_1$) of 15(ft$_3$) per (second$_2$), from a fissure that opened in mid-December.
The Italian Army (yesterday$_2$) detonated 400lb of dynamite 3,500 feet up Mount Etna's slopes.

[Mrs Allan] was taken to nearby Kelowna General Hospital after the (body$_1$) was found.[Her] husband, Stuart, 52, said yesterday he had been in daily contact with [her] since [she] flew to Canada last (month$_2$) on the second pilgrimage to find [her] son."[She] is suffering from exhaustion but otherwise fine," he said."I spoke to [her] last (night$_2$) and [she] is under strict orders to have complete rest."

A (policeman$_1$) was yesterday jailed for seven years for raping an 18-year-old (woman$_2$) in his marked patrol car while he was on duty and in uniform. Sentencing constable [Peter Anderson], 41, Mr Justice Jowitt told him he had done "great damage to the trust in (police$_1$)". [Anderson], married with two children, attacked the (woman$_2$) in a deserted [allotment], after agreeing to give her and (boyfriend$_2$) a lift home from a discotheque. He first dropped the (man$_2$) off and then drove to the [allotment].

Figure 2
Excerpts of documents from our test set with discourse annotations. Centers are in double boxes; terms occurring in lexical chains are in oval boxes. Words with the same subscript are members of the same chain (e.g., *police*, *policeman*).

different properties. Specifically, although each utterance may contain several entities, it is assumed that a *single entity* is "centered," thereby representing the current discourse focus. One of the main claims underlying centering is that discourse segments in which successive utterances contain common centers are more coherent than segments where the center repeatedly changes.

Each utterance U_j in a discourse has a list of **forward-looking centers**, $C_f(U_j)$, and a *unique* **backward-looking center**, $C_b(U_j)$. $C_f(U_j)$ represents a ranking of the entities invoked by U_j according to their salience. Thus, some entities in the discourse are deemed more important than others. The C_b of the current utterance U_j is the highest-ranked element in $C_f(U_{j-1})$ that is also in U_j. (Centering hypothesizes that the C_b is likely to be realized as a pronoun.) Entities are commonly ranked in terms of their grammatical function, namely, subjects are ranked more highly than objects, which are more highly ranked than the rest (Grosz, Weinstein, and Joshi 1995). The C_b links U_j to the previous discourse, but it does so *locally* since $C_b(U_j)$ is chosen from U_{j-1}.

Centering formalizes fluctuations in topic continuity in terms of transitions between adjacent utterances. Grosz, Weinstein, and Joshi (1995) distinguish between three

types of transitions. In CONTINUE transitions, $C_b(U_j) = C_b(U_{j-1})$ and $C_b(U_j)$ is the most highly ranked element entity in U_j. In RETAIN transitions $C_b(U_j) = C_b(U_{j-1})$ but $C_b(U_j)$ is not the most highly ranked element entity in U_j. And in SHIFT transitions $C_b(U_j) \neq C_b(U_{j-1})$. These transitions are ordered: CONTINUEs are preferred over RETAINs, which are preferred over SHIFTs. And discourses with many CONTINUE transitions are considered more coherent than those which repeatedly SHIFT from one center to the other.

We demonstrate these concepts in passages (1a)–(1c) taken from Walker, Joshi, and Prince (1998).

(1) a. Jeff helped Dick wash the car.
`CF(Jeff, Dick, car)`
b. He washed the windows as Dick waxed the car.
`CF(Jeff, Dick, car)`
`CB=Jeff`
c. He soaped a pane.
`CF(Jeff, pane)`
`CB=Jeff`

Here, the first utterance does not have a backward-looking center but has three forward-looking centers *Jeff*, *Dick*, and *car*. To determine the backward-looking center of (1b) we find the highest ranked entity among the forward-looking centers in (1a) which also occurs in (1b). This is *Jeff* as it is the subject (and thus most salient entity) in (1a) and present (as a pronoun) in (1b). The same procedure is applied for utterance (1c). Also note that (1a) and (1b) are linked via a CONTINUE transition. The same is true for (1b) and (1c).

For the purposes of our document compression application, we are not so much interested in characterizing our texts in terms of entity transitions. Because they are all written by humans, we can assume they are more or less coherent. Nonetheless, identifying the centers in discourse seems important. These will indicate what the document is about, who the main protagonists are, and how the discourse focus progresses. We would probably not want to delete entities functioning as backward-looking centers.

As Centering is primarily a linguistic theory rather than a computational one, it is not explicitly stated how the concepts of "utterance," "entities," and "ranking" are instantiated. A great deal of research has been devoted to fleshing these out and many different instantiations have been developed in the literature (see Poesio et al. [2004] for details). In our case, the instantiation will have a bearing on the reliability of the algorithm to detect centers. If the parameters are too specific then it may not be possible to accurately determine the center for a given utterance. Because our aim is to identify centers in discourse automatically, our parameter choice is driven by two considerations: robustness and ease of computation.

We therefore follow previous work (e.g., Miltsakaki and Kukich 2000) in assuming that the unit of an utterance is the sentence (i.e., a main clause with accompanying subordinate and adjunct clauses). This is a simplistic view of an utterance; however it is in line with our compression task, which also operates over sentences. We determine which entities are invoked by a sentence using two methods. First, we perform named entity identification and coreference resolution on each document using LingPipe,[6] a

6 LingPipe can be downloaded from `http://alias-i.com/lingpipe/`.

publicly available system. Named entities are not the only type of entity to occur in our data, thus to ensure a high entity recall we add named entities and all remaining nouns[7] to the C_f list. Entity matching between sentences is required to determine the C_b of a sentence. This is done using the named entity's unique identifier (as provided by LingPipe) or by the entity's surface form in the case of nouns not classified as named entities.

We follow Grosz, Weinstein, and Joshi (1995) in ranking entities according to their grammatical roles; subjects are ranked more highly than objects, which are in turn ranked higher than other grammatical roles; ties are broken using left-to-right ordering of the grammatical roles in the sentence (Tetreault 2001). We identify grammatical roles using RASP (Briscoe and Carroll 2002). Formally, our centering algorithm is as follows (where U_j corresponds to sentence j):

1. Extract entities from U_j.
2. Create $C_f(U_j)$ by ranking the entities in U_j according to their grammatical role (subjects $>$ objects $>$ others, ties broken using left-to-right word order of U_j).
3. Find the highest ranked entity in $C_f(U_{j-1})$ which occurs in $C_f(U_j)$; set the entity to be $C_b(U_j)$.

This procedure involves several automatic steps (named entity recognition, coreference resolution, and identification of grammatical roles) and will unavoidably produce some noisy annotations. There is no guarantee, therefore, that the right C_b will be identified or that all sentences will be marked with a C_b. The latter situation also occurs in passages that contain abrupt changes in topic. In such cases, none of the entities realized in U_j will occur in $C_f(U_{j-1})$. Hopefully, lexical chains will come to the rescue here as an alternative means of capturing local content within a document.

Figure 2 shows the centers (in double boxes) identified by our algorithm. In the first document *lava* and *debris* are marked as centers, in the second document *Mrs Allan* (and its coreferents), and in the third one *Peter Anderson* and *allotment*. When comparing the annotations produced by centering and the lexical chains, we observe that they tend to be complementary. Proper nouns that lexical chains miss out on are often identified by centering. When the latter fails, due to errors in coreference resolution or the identification of grammatical relations, lexical chains can be more robust because only WordNet is required for their computation. As an example consider the third document in Figure 2. Here, lexical chains provide a better insight into the text. Were we to rely solely on centering, we would obtain annotations only for two entities, namely, *Peter Anderson* and *allotment*.

5. The Discourse-Inspired Compression Model

We now turn our attention to incorporating discourse information into our compression model. Before compression takes place, all documents are processed using the centering and lexical chain algorithms described earlier. In each sentence we annotate the center $C_b(U_j)$ if one exists. Words (or phrases) that are present in the current sentence and function as the center in the next sentence $C_b(U_{j+1})$ are also flagged. Finally, words

7 As determined by the word's part-of-speech tag.

are marked if they are part of a prevalent (high scoring) chain. Provided with this additional knowledge our model takes a (sentence-separated) source document as input and generates a compressed version by applying sentence-level and discourse-level constraints to the *entire* document rather than to each sentence sequentially. In our earlier formulation of the compression task (Clarke and Lapata 2008), we create and solve an ILP for every sentence, whereas now an ILP is solved for each document. This makes sense from a discourse perspective as compression decisions are not made independently of each other. Also note that this latter formulation brings compression closer to summarization as we can manipulate the document compression rate directly, for example, by adding a constraint that forces the target document to be less than b tokens. This allows the model to choose how much to compress each individual sentence without requiring that they all have the same compression rate. Accordingly, we modify our objective function by introducing a sum over all sentences (assuming l sentences are present in the document) and adding an additional index g to each decision variable to track the sentence it came from:

$$\begin{aligned}\max z = \sum_{g=1}^{l} \Bigg[& \sum_{i=1}^{n_g} \delta_{g,i} \cdot \lambda I(x_{g,i}) + \sum_{i=1}^{n_g} \alpha_{g,i} \cdot P(x_{g,i}|\text{start}) \\ & + \sum_{i=1}^{n_g-2} \sum_{j=i+1}^{n_g-1} \sum_{k=j+1}^{n_g} \gamma_{g,ijk} \cdot P(x_{g,k}|x_{g,i}, x_{g,j}) \\ & + \sum_{i=0}^{n_g-1} \sum_{j=i+1}^{n_g} \beta_{g,ij} \cdot P(\text{end}|x_{g,i}, x_{g,j}) \Bigg] \\ & - \zeta_{min} \cdot \mu - \zeta_{max} \cdot \mu \end{aligned} \tag{18}$$

We also modify the compression rate soft constraint to act over the whole document rather than sentences. This allows some sentences to violate the compression rate without incurring a penalty, provided the compression rate of the document falls within the specified range.

Document Compression Rate Constraints. We wish to penalize compressions which do not fall within a desired compression rate range ($c_{min}\% - c_{max}\%$).

$$\sum_{g=1}^{l} \sum_{i=0}^{n_g} \delta_{g,i} + \zeta_{min} \geq c_{min} \cdot \sum_{g=1}^{l} n_g \tag{19}$$

$$\sum_{g=1}^{l} \sum_{i=0}^{n_g} \sum_{i=0}^{n_g} \delta_{g,i} - \zeta_{max} \leq c_{max} \cdot \sum_{g=1}^{l} n_g \tag{20}$$

Besides the new objective function and compression rate constraints, the model makes use of all the sentence-level constraints introduced in Section 3.3, but is crucially enhanced with three discourse constraints explained in the following.

5.1 Discourse Constraints

Our first goal to is preserve the focus of each sentence. If the center, C_b, is identified in the source sentence it must be retained in the target compression. If present, the entity realized as the C_b in the following sentence should also be retained to ensure the focus is preserved from one sentence to the next. Such a condition is easily captured with the following ILP constraint:

$$\delta_i = 1 \quad \forall i : x_i \in \{C_b(U_j), C_b(U_{j+1})\} \tag{21}$$

As an example, consider the first discourse in Figure 2. The constraints generated from Equation (21) will require the compression to retain *lava* in the first two sentences and *debris* in the second and third sentences.

As mentioned in the previous section, the centering algorithm relies on NLP technology that is not 100% accurate (named entity detection, parsing, and coreference resolution). Therefore, the algorithm can only approximate the center for each sentence and in some cases fails to identify any centers at all. Lexical chains provide a complementary annotation of the topic or theme of the document using information which is not restricted to adjacent sentences. Recall that once chains are created, they are scored, and chains with scores less than the average are discarded. We consider all remaining lexical chains as topical and require that words in these be retained in the compression.

$$\delta_i = 1 \quad \forall i : x_i \in \text{document topical lexical chain} \tag{22}$$

Consider again the first text in Figure 2. Here, *flow* and *rate* are members of the same chain (marked with subscript 1). According to constraint (22) both words must be included in the compressed document. In the third document the words relating to "police" (*police*, *policeman*) and "people" (*woman*, *boyfriend*, *man*) also would be retained in the compression.

Our final discourse constraint concerns pronouns. Specifically, we force personal pronouns (whose antecedent may not always be identified) to be included in the compression.

$$\delta_i = 1 \quad \forall i : x_i \in \text{personal pronouns} \tag{23}$$

The constraints just described ensure that the compressed document will retain the discourse flow of the source document and will preserve terms indicative of important topics. Document compression aside, the discourse constraints will also benefit sentence-level compression. They provide our model, which so far relied on syntactic evidence and surface level document characteristics (i.e., word frequencies), additional evidence for retaining (discourse) relevant words.

5.2 Applying the Constraints

As explained earlier we apply the model and the constraints to each document. In our earlier sentence-based formulation, a significance score (see Section 3.2) was used to highlight which nouns and verbs should be included in the compression. As far as nouns are concerned, our discourse constraints perform a similar task. Thus, when a sentence contains discourse annotations, we are inclined to trust them more and only calculate the significance score for verbs.

During development it was observed that applying all discourse constraints simultaneously (see Equations (21)–(23)) results in relatively long compressions. To counteract this, we employ these constraints using a back-off strategy that relies on progressively less reliable information. Our back-off model works as follows: If centering information is present, we apply the appropriate constraints (Equation (21)). If no centers are present, we back off to the lexical chain information using Equation (22), and in the absence of the latter we back off to the pronoun constraint (Equation (23)). Finally, if discourse information is entirely absent from the sentence, we default to the significance score. Sentential constraints are applied throughout irrespective of discourse constraints. We determined this ordering (i.e., centering first, then lexical chains, and then pronouns) on the development set. Centering tends to be more precise, whereas lexical chains have high recall but lower precision in terms of identifying which entities are in focus and should therefore not be dropped. In our test data (see Section 6 for details), the centering constraint was used in 68.6% of the sentences. The model backed off to lexical chains for 13.7% of the test sentences, whereas the pronoun constraint was applied in 8.5%. Finally, the noun and verb significance score was used on the remaining 9.2%. Examples of our system's output for the texts in Figure 2 are given in Figure 3.

6. Experimental Set-up

In this section we present our experimental set-up for assessing the performance of the compression model. We describe the compression corpus used in our study, briefly introduce the model used for comparison with our approach, and explain how system output was evaluated.

6.1 Compression Corpus

Previous work on sentence compression has used almost exclusively the Ziff-Davis corpus, a compression corpus derived automatically from document–abstract pairs (Knight and Marcu 2002). Unfortunately, this corpus is not suitable for our purposes because it consists of isolated sentences taken from several different documents. We thus created a document-based compression corpus manually. Specifically, annotators were presented with one document at a time and asked to compress sentences sequentially by removing tokens. They were free to remove any words they deemed superfluous, provided their deletions (a) preserved the most important information in the source sentence, and (b) ensured the compressed sentence remained grammatical. If they wished, they could leave a sentence uncompressed. They were not allowed to delete whole sentences even if they believed they contained no information content with respect to the story, as this would blur the task with summarization. Following these guidelines,

Bad weather dashed hopes ~~of attempts~~ to halt the flow during what was seen as ~~a~~ lull in lava's momentum. Experts say that even if the eruption stopped ~~today,~~ the pressure of lava piled ~~up behind for six miles~~ would bring debris cascading ~~down on to the town anyway~~. Some estimate volcano is pouring ~~out one~~ million tons of debris ~~a day, at a rate of 15ft per second,~~ from ~~a~~ fissure ~~that~~ opened in mid-December. ~~The Italian~~ Army yesterday detonated ~~400~~ lb ~~of dynamite 3,500 feet up Mount Etna's slopes~~.

Mrs Allan was taken to ~~nearby~~ Kelowna ~~General~~ Hospital ~~after the body was found~~. Her husband, Stuart, ~~52,~~ said ~~yesterday~~ he had been in ~~daily~~ contact with her since she flew ~~to Canada~~ last month to find her son. "She is suffering ~~from exhaustion but otherwise fine,~~" he said. "I spoke to her last night and she is under ~~strict~~ orders to have ~~complete~~ rest."

~~A~~ policeman was ~~yesterday~~ jailed ~~for seven years~~ for raping an ~~18-year-old~~ woman ~~in his marked patrol car~~ while he was on duty and in uniform. ~~Sentencing constable~~ Peter Anderson, Jowitt told him he had done "~~great~~ damage to ~~the~~ trust ~~in the police~~". Anderson, married with ~~two~~ children, attacked the woman in ~~deserted~~ allotment after agreeing to give her and boyfriend a lift home ~~from a discotheque~~. He ~~first~~ dropped ~~the~~ man ~~off~~ and ~~then~~ drove to ~~the~~ allotment.

Figure 3
Compression output on excerpts from Figure 2 using the discourse model. Words that are dropped are striken out.

the annotators created compressions for 82 stories (1,629 sentences) from the BNC and the LA Times and Washington Post.[8] Forty-eight (48) documents (962 sentences) were used for training, 3 for development (63 sentences), and 31 for testing (604 sentences).

6.2 Comparison with State-of-the-Art

The discourse-based compression model was evaluated against our earlier sentence-based ILP model (without the discourse constraints). In addition, we compared our approach against a state-of-the-art model which does not take discourse-level information into account, does not use ILP, and is sentence-based. We give a brief description in the following, and refer the interested reader to McDonald (2006) for details.

McDonald (2006) formalizes sentence compression as a classification task in a discriminative large-margin learning framework: Pairs of words from the source sentence are classified as being adjacent or not in the target compression. Let $\mathbf{x} = x_1, \ldots, x_n$ denote a source sentence with a target compression $\mathbf{y} = y_1, \ldots, y_m$ where each y_j occurs in $\mathbf{x}$. The function $L(y_i) \in \{1 \ldots n\}$ maps word y_i in the target to the index of the word in the source, $\mathbf{x}$ (subject to the constraint that $L(y_i) < L(y_{i+1})$). McDonald defines the score of a compression $\mathbf{y}$ for a sentence $\mathbf{x}$ as the dot product between

8 The corpus is available from `http://homepages.inf.ed.ac.uk/s0460084/data/`.

a high-dimensional feature representation over bigrams and a corresponding weight vector:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{j=2}^{|\mathbf{y}|} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, L(y_{j-1}), L(y_j)) \tag{24}$$

Decoding in this framework amounts to finding the combination of bigrams that maximize the scoring function in Equation (24). The maximization is solved using dynamic programming (see McDonald [2006] for details).

The model parameters are estimated using the Margin Infused Relaxed Algorithm (MIRA; Crammer and Singer 2003), a discriminative large-margin online learning technique. This algorithm learns by compressing each sentence and comparing the result with the gold standard. The weights are updated so that the score of the correct compression (the gold standard) is greater than the score of all other compressions by a margin proportional to their loss. The loss function is the number of words falsely retained or dropped in the incorrect compression relative to the gold standard. McDonald employs a rich feature set defined over words, parts of speech, phrase structure trees, and dependencies. These are gathered over adjacent words in the compression and the words in between which were dropped.

It is important to note that McDonald (2006) is not a straw-man system. It achieves highly competitive performance compared with Knight and Marcu's (2002) noisy-channel and decision-tree models. Due to its discriminative nature, the model is able to use a large feature set and to optimize compression accuracy directly. In other words, McDonald's model has a head start against our own model which does not utilize a large parallel corpus and has only a few constraints. The comparison of the two systems allows us to establish that we have a competitive state-of-the-art system, even without discourse constraints.

We trained McDonald's (2006) model on the full training set (48 documents, 962 sentences). Our implementation used an identical feature set, the only difference being that our phrase structure and dependency features were extracted from the output of Roark's (2001) parser. McDonald uses Charniak's (2000) parser, which performs comparably. We also employed a slightly modified loss function to encourage compression on our data set. McDonald's results were reported on the Ziff-Davis corpus. The language model required for the ILP system was trained on 80 million tokens from the English GigaWord corpus (LDC2007T07) using the SRI Language Modeling Toolkit with Kneser-Ney discounting. The significance score was calculated on 80 million tokens from the same corpus. The ILP model presented in Equation (1) implements a weighted combination of the significance score with a language model. The weight was tuned on the development set which consisted of three source documents and their target compressions. Our optimization procedure used Powell's method (Press et al. 1992) and a loss function based on the grammatical relations F1 between the gold standard and system output. The optimal weight was approximately 9.0. Note that the development set was the only source of parallel data our model had access to.

In order to compare all three models (sentence-based ILP, discourse-based ILP, and McDonald [2006]) on an equal footing, we ensured that their compression rates were similar. To do this, we first run McDonald's model on our data and then set the compression rate for our ILP models so that it is comparable to his output. This can be done relatively straightforwardly by adjusting the compression rate range soft constraint. In our experiments we set the minimum compression rate to 57%, the upper rate to 62%,

and the violation penalty (μ) to -99. In practice, the soft constraint controlling the compression rate can be removed or specifically tuned to suit the application.

6.3 Evaluation

Previous studies evaluate the well-formedness of automatically generated compressions out of context. The target sentences are typically rated by naive subjects on two dimensions, grammaticality and importance (Knight and Marcu 2002). Automatic evaluation measures have also been proposed. Riezler et al. (2003) compare the grammatical relations found in the system output against those found in a gold standard using F1. Although F1 conflates grammaticality and importance into a single score, it nevertheless has been shown to correlate reliably with human judgments (Clarke and Lapata 2006).

The aims of our evaluation study were twofold. Firstly, we wanted to examine whether our discourse constraints improve the compressions for individual sentences. There is no hope for generating shorter documents if the compressed sentences are either too wordy or too ungrammatical. Secondly and more importantly, our goal was to evaluate the compressed documents as a whole by examining whether they are readable and the degree to which they retain key information when compared to the originals. We evaluated sentence-based compressions automatically using F1 and the grammatical relations annotations provided by RASP (Briscoe and Carroll 2002). This parser is suited to the compression task as it provides parses for both full sentences and sentence fragments and is generally robust enough to analyze semi-grammatical sentences. We computed F1 over all the relations provided by RASP (e.g., subject, direct/indirect object, modifier; 17 in total). We compared the output of our discourse system on the test set (31 documents, 604 sentences) against the sentence-based ILP model and McDonald (2006).

Our document-level evaluation was motivated by two questions: (1) Are the compressed documents readable? and (2) How much key information is preserved between the source document and its target compression? The readability of a document is fairly straightforward to measure by asking participants to provide a rating (e.g., on a seven-point scale). Measuring how much information is preserved in the compressed document is more involved. Under the assumption that the target document is to function as a replacement for the source, we can measure the extent to which the compressed version can be used to find answers for questions which have been derived from the source and are representative of its core content. We thus created questions from the source and then determined whether it was possible to find their answers by reading the compressed target. The more questions a hypothetical compression system can answer, the better it is at compressing the document as a whole.

A question-answering (Q&A) paradigm has been used previously to evaluate summaries and text compression. Morris, Kasper, and Adams (1992) performed one of the first Q&A evaluations to investigate the degree to which documents could be summarized before reading comprehension diminished. Their corpus consisted of four passages randomly selected from a set of sample Graduate Management Aptitude Test (GMAT) reading comprehension tests. The texts covered a range of topics including medieval literature, 18th-century Japan, minority-operated businesses, and Florentine art. Accompanying each text were eight multiple-choice questions, each containing five possible answers. The questions were provided by the Educational Testing Service and were designed to measure the subjects' reading comprehension. Subjects were

given various textual treatments: the full text, a human-authored abstract, three system-generated extracts, and a final treatment where merely the questions were presented without any text. The questions-only treatment was used as a control to investigate if subjects could answer questions without any source material. Subjects were instructed to read the passage (if provided) and answer the multiple choice questions.

The advantage of using standardized tests, such as the GMAT reading comprehension test, is that Q&A pairs are provided along with a method for scoring answers (the correct answer is one among five possible choices). However, our corpora do not contain ready prepared Q&A pairs; thus we require a methodology for constructing questions and their answers and scoring documents against the answers. One such methodology is presented in the TIPSTER Text Summarization Evaluation (SUMMAC; Mani et al. 2002). SUMMAC was concerned with producing summaries tailored to specific topics. The Q&A task involved an evaluation where a topic-related summary for a document was evaluated in terms of its "informativeness," namely, the degree to which it contained answers found in the source document to a set of topic-related questions. For each topic (three in total), 30 relevant documents were chosen to generate a single summary. One annotator per topic came up with no more than five questions relating to the **obligatory aspects** of the topic. An obligatory aspect of a topic was defined as information that must be present in the document for the document to be relevant to the topic. The annotators then created an answer key for their topic by annotating the passages and phrases from the documents which provided the answers to the questions. In the SUMMAC evaluation, the annotator for each topic was tasked with scoring the system summaries. Scoring involved comparing the summaries against the answer key (annotated passages from the source documents) while judging whether the summary provided a *Correct*, *Partially Correct*, or *Missing* answer. If a summary contained an answer key and sufficient context the summary was deemed correct; however, summaries would be considered partially correct if the answer key was present but with insufficient context. If context was completely missing, misleading, or the answer key was absent then the summary was judged missing.

Our methodology for constructing Q&A pairs and for scoring documents is inspired by the SUMMAC evaluation exercise (Mani et al. 2002). Rather than creating questions for document sets (or topics) our questions were derived from individual documents. Two annotators were independently instructed to read the documents from our (test) corpus and create Q&A pairs. Each annotator drafted no more than ten questions and answers per document, related to its content. Annotators were asked to create fact-based questions which required an unambiguous answer; these were typically who, what, where, when, and how–style questions. The purpose of using two annotators per document was to allow annotators to compare and revise their Q&A pairs; this process was repeated until a common agreed-upon set of questions was reached. Revisions typically involved merging and simplifying questions to make them clearer, and in some cases splitting a question into multiple questions. Documents for which too few questions were agreed upon and for which the questions and answers were too ambiguous were removed. This left an evaluation set of six documents with between five to eight concise questions per document. Figure 4 shows a document from our test set and the questions and answers our annotators created for it.

For scoring our documents we adopt a more objective method than SUMMAC. Instead of asking the annotator who constructed the questions to check the document compressions for the answers, we ask naive participants to read the compressed documents and answer the questions as best as they can. During evaluation, the source document is not shown to our subjects; thus, if the compression is difficult to read, the

Snow, high winds and bitter disagreement yesterday further hampered attempts to tame Mount Etna, which is threatening to overrun the Sicilian town of Zafferana with millions of tons of volcanic lava.

The wall of molten lava has come to a virtual halt 150 yards from the first home in the town, but officials said yesterday that its flow appeared to have picked up speed further up the slope. A crust appears to have formed over the volcanic rubble, but red-hot lava began creeping over it yesterday and into a private orchard. Bad weather dashed hopes of attempts to halt the flow during what was seen as a natural lull in the lava's momentum.

Some experts say that even if the eruption stopped today, the sheer pressure of lava piled up behind for six miles would bring debris cascading down on to the town anyway. Some estimate the volcano is pouring out one million tons of debris a day, at a rate of 15ft per second, from a fissure that opened in mid-December.

The Italian army yesterday detonated nearly 400lb of dynamite 3,500 feet up Mount Etna's slopes. The explosives, which were described as nothing more than an experiment, were detonated just above a dam built in January and breached last week. They succeeded in closing off the third of five underground conduits formed beneath the surface crust and through which red-hot magma has been flowing. But the teams later discovered that the conduit was dry, suggesting that the lava had already found a new course.

Rumours have been circulating that experts are bitterly divided over what to do. But in another experiment 50 two-ton concrete slabs are to be chained together and dumped from a huge tilting steel platform about 6,750ft above sea level. It is hoped the slabs will block the conduit from which the main force of the lava is said to be bearing down "like a train", causing it to break up and cool. High winds and snowfalls have, however, grounded at a lower level the powerful US Navy Sea Stallion helicopters used to transport the slabs.

Prof Letterio Villari, a noted vulcanologist, said yesterday he had "absolutely no faith whatsoever" in the plan. If Zafferana was saved from the lava, which could flow for a year or more, it would be "a complete fluke", he said.

Question	Answer
What is posing a threat to the town?	lava
What hindered attempts to stop the lava flow?	bad weather
What are the Army attempting to block to halt the lava flow?	underground conduits
What did the Army do first to stop the lava flow?	detonate explosives
What other experiments are planned?	using concrete slabs
Do the experts agree over what to do next?	no

Figure 4
Example document from our test set and questions with answer key created for this document.

participants have no point of reference to help them understand the compression. This is a departure from previous evaluations within text generation tasks, where the source text is available at judgment time; in our case only the system output is available.

The document-based evaluation was conducted remotely over the Internet using a custom-built Web interface. Upon loading the Web interface, participants were presented with a set of instructions that explained the Q&A task and provided examples.

Table 1
Compression results: compression rate and relation-based F1.

Model	CompR	Precision	Recall	F1
McDonald	60.1%	43.9%	36.5%*	37.9%*
Sentence ILP	62.1%	40.7%*	39.4%*	39.0%*
Discourse ILP	61.0%	46.2%	44.2%	42.2%
Gold Standard	70.3%	—	—	—

* Significantly different from Discourse ILP ($p < 0.01$ using the Wilcoxon test).

Subjects were first asked to read the compressed document and then rate its readability on a seven-point scale where 7 = excellent, and 1 = terrible. Next, questions were presented one at a time (the order being is defined by the annotators) and participants were encouraged to consult the document for the answer. Answers were written directly into a text field on the Web interface which allowed free-form text to be submitted. Once a participant provided an answer and confirmed the answer, the interface locked the answer to ensure it was not modified later. This was necessary because later questions could reveal information which would help answer previous questions.

We elicited answers for six documents in four compression conditions: gold standard, using the ILP sentence-based model, the ILP discourse model, and McDonald's (2006) model. A Latin square design was used to prevent participants from seeing multiple treatments (compressions) of the same document thus removing any learning effect. A total of 116 unpaid volunteers completed the experiment. They were recruited through student mailing lists and the Language Experiments Web site.[9] The answers provided by our subjects were scored against an answer key. A correct answer was marked with a score of one, and zero otherwise. In cases where two answers were required, a score of 0.5 was awarded to each correct answer. The score for a compressed document is the average of its question scores. All subsequent tests and comparisons are performed on the document score.

7. Results

We first assessed the compressions produced by the two ILP models (Discourse and Sentence) and McDonald (2006) on a sentence-by-sentence basis. Table 1 shows the compression rates (CompR) for the three systems and evaluates the quality of their output using grammatical relations F1. As can be seen, all three systems produce comparable compression rates. The Discourse ILP compressions are slightly longer than McDonald's (2006) (61.0% vs. 60.1%) and slightly shorter than the Sentence ILP model (61.0% vs. 62.1%). The Discourse ILP model is significantly better than McDonald (2006) and Sentence ILP in terms of F1, indicating that discourse-level information is generally helpful. All three systems could use further improvement, as inter-annotator agreement on this data yields an F1 of 65.8% (Clarke 2008).

Let us now consider the results of our document-based evaluation. Table 2 shows the mean readability ratings obtained for each system and the percentage of questions answered correctly. We used an analysis of variance (ANOVA) to examine the effect

9 Available at `http://www.language-experiments.org`.

Table 2
Human evaluation results: average readability ratings and average percentage of questions answered correctly.

Model	Readability	Q&A (%)
McDonald	2.52*	51.42*†
Sentence ILP	2.76*	52.35*†
Discourse ILP	3.10*	71.38*
Gold Standard	5.41†	85.48†

* Significantly different from Gold Standard.
† Significantly different from Discourse ILP.

of compression type (McDonald, Sentence ILP, Discourse ILP, Gold Standard). The ANOVA revealed a reliable effect on both readability and Q&A. Post hoc Tukey tests showed that McDonald and the two ILP models do not differ significantly in terms of readability. However, they are all significantly less readable than the gold standard ($\alpha < 0.01$). For the Q&A task, we observe that our system is significantly better than McDonald ($\alpha < 0.01$) and Sentence ILP ($\alpha < 0.01$), but significantly worse than the gold standard ($\alpha < 0.05$). McDonald and Sentence ILP yield comparable performance (their difference is not statistically significant).

These results indicate that the automatic systems lag behind the human gold standard in terms of readability. When reading entire documents, subjects are less tolerant of ungrammatical constructions. We also find out that, despite relatively low readability, the documents are overall understandable. The discourse-based model generates more informative documents—the number of questions answered correctly increases by 19% in comparison to McDonald and Sentence ILP. This is an encouraging result suggesting that there are advantages in developing compression models that exploit discourse-level information information.

Figure 5 shows the output of the ILP systems (Discourse and Sentence) on two test documents. Words that are dropped have been stricken out. As can be seen, the two systems produce different compressions, and the discourse-based output is more coherent. This is corroborated by the readability results where the discourse ILP model received the highest rating. Also note that some of the compressions produced by the sentence-based model distort the meaning of the original text, presumably leading the reader to make wrong inferences. For example, in the second document (Sentence ILP version) one infers that the victim was urged to report the incident. Moreover, important information is often omitted, for example, that the victim was indeed raped or that the strike would be damaging not only to the company but also to its staff (see the Sentence ILP version in the first document).

8. Conclusions and Future Work

In this article we proposed a novel method for automatic sentence compression. Central in our approach is the use of discourse-level information, which we argue is an important prerequisite for document (as opposed to sentence) compression. Our model uses integer linear programming for inferring globally optimal compressions in the presence of linguistically motivated constraints. Our discourse constraints aim to capture local coherence and are inspired by Centering Theory and lexical chains. We showed that our

Discourse ILP	Improvements ~~in certain allowances~~ were made, described as divisive by ~~the~~ unions, ~~but~~ the company has refused to compromise ~~on a reduction in the shorter working week~~. Ford dismissed ~~an immediate~~ meeting with the unions ~~but~~ did not rule out talks after Christmas. It said that a strike would be damaging to the company and to its staff. Production closed ~~down~~ at ~~Ford last~~ night for the ~~Christmas~~ period. Plants will open again on January ~~2~~.
Sentence ILP	Improvements in ~~certain~~ allowances were made~~,~~ described ~~as divisive by the unions, but~~ the company has refused to compromise on a reduction in the ~~shorter working week~~. Ford dismissed ~~an immediate~~ meeting with ~~the~~ unions ~~but did~~ not rule ~~out~~ talks after Christmas. It said that ~~a~~ strike would be damaging to ~~the~~ company ~~and to its staff~~. Production closed ~~down~~ at Ford ~~last~~ night for ~~the Christmas period~~. Plants ~~will~~ open again on January ~~2~~.

Discourse ILP	He threatened her by forcing his truncheon under her chin and ~~then~~ raped her. She said he ~~only~~ refrained from inserting his truncheon into her~~,~~ after she begged him ~~not to~~. ~~Afterwards~~ he told her not to report ~~the~~ incident because he could have her "nicked" for soliciting. She did not report it because she did not think she would be believed. Police investigated after ~~an anonymous~~ report.
Sentence ILP	He threatened her by forcing his truncheon under her chin ~~and then raped her~~. She said he ~~only~~ refrained from inserting his truncheon into her~~, after she begged him not to~~. ~~Afterwards~~ he told her ~~not to~~ report ~~the~~ incident because he ~~could~~ have her "nicked" for soliciting . ~~She did not report it~~ because she did not think she would be believed. Police investigated after ~~an anonymous~~ report.

Figure 5
Output of Discourse and Sentence ILP systems on two test documents. Words that are stricken out have been dropped.

model can be successfully employed to produce compressed documents that preserve most of the original core content.

Our results confirm the conventional wisdom that discourse-level information is helpful in summarization. We also show that this type of information can be identified robustly in free text. Our experiments focused primarily on local discourse structure using two complementary representations. Centering tends to produce more annotations since it tries to identify a center in every sentence. Lexical chains tend to provide more general information, such as the major topics in a document. Due to their approximate nature, there is no one representation that is uniquely suited to the compression task. Rather, it is the synergy between lexical chains and centering that brings improvements. The discourse annotations proposed here are not specific to our model. They could be easily translated into features and incorporated into discriminative modeling paradigms (e.g., Nguyen et al. 2004; McDonald 2006; Cohn and Lapata 2009). The same is true for the Q&A evaluation paradigm employed in our experiments. It could be straightforwardly adapted to assess the information content of shorter summaries and potentially used to perform large-scale comparisons within and across systems.

Our approach differs from most summarization work in that our summaries are fairly long. However, we believe this is the first step to understanding how compression can help summarization. An obvious extension would be to interface our

compression model with sentence extraction (see Martins and Smith [2009] for an ILP formulation of a model that jointly performs sentence extraction and compression, without, however, taking discourse level information into account). The discourse annotations can help guide the extraction method into selecting topically related sentences which can consequently be compressed together. More generally, formulating the summarization process in the ILP framework outlined here would allow the integration of varied and sometimes conflicting constraints during summary generation. Examples include the summary length, and whether it is coherent, grammatical, or repetitive. Additional flexibility can be introduced by changing some of the constraints from hard to soft (as we did with the compression rate constraints), although determining the penalty for constraint violation manually using prior knowledge is a non-trivial task (Chang, Ratinov, and Roth 2007) and automatically learning the constraint penalty results in a harder learning problem. Importantly, under the ILP formulation such constraints can be explicitly encoded and applied during inference while finding a globally optimal solution.

Acknowledgments

We are grateful to Ryan McDonald for his help with the re-implementation of his system, and our annotators Vasilis Karaiskos and Sarah Luger. Thanks to Alex Lascarides, Sebastian Riedel, and Bonnie Webber for insightful comments and suggestions, and to the anonymous referees whose feedback helped to substantially improve the present article. Lapata acknowledges the support of EPSRC (grant GR/T04540/01).

References

Aho, A. V. and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3:37–56.

Barzilay, R. and M. Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL-97 Intelligent Scalable Text Summarization Workshop*, pages 10–17, Madrid.

Barzilay, Regina and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 359–366, New York, NY.

Barzilay, Regina and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Boguraev, Branimir and Chris Kennedy. 1997. Salience-based content characterization of text documents. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid.

Brin, Sergey and Michael Page. 1998. Anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th Conference on World Wide Web*, pages 107–117, Brisbane.

Briscoe, E. J. and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC–2002)*, pages 1499–1504, Las Palmas.

Carlson, Lynn, John M. Conroy, Daniel Marcu, Dianne P. O'Leary, Mary E. Okurowski, and Anthony Taylor. 2001. An empirical study on the relation between abstracts, extracts, and the discourse structure of texts. In *Proceedings of the DUC-2001 Workshop on Text Summarization*, New Orleans, LA.

Chang, Ming-Wei, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *Proceedings of the 22nd International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 280–287, Prague.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American Annual Meeting of the Association for Computational Linguistics*, pages 132–139, Seattle, WA.

Clarke, James. 2008. *Global Inference for Sentence Compression: An Integer Linear Programming Approach*. Ph.D. thesis, University of Edinburgh.

Clarke, James and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st International*

Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 377–384, Sydney.

Clarke, James and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Cohn, Trevor and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Corston-Oliver, Simon. 2001. Text compaction for display on very small screens. In *Proceedings of the NAACL Workshop on Automatic Summarization*, pages 89–98, Pittsburgh, PA.

Corston-Oliver, Simon H. 1998. Computing representations of the structure of written discourse. Technical Report MSR-TR-98-15, Microsoft Research, Redmond, WA.

Crammer, Koby and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Daumé III, Hal and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 449–456, Philadelphia, PA.

Denis, Pascal and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 236–243, Rochester, NY.

Dras, Mark. 1997. Reluctant paraphrase: Textual restructuring under an optimisation model. In *Proceedings of the Fifth Biannual Meeting of the Pacific Association for Computational Linguistics*, pages 98–104, Ohme.

Endres-Niggemeyer, Brigitte. 1998. *Summarising Information*. Springer, Berlin.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Galley, Michel and Kathleen McKeown. 2003. Improving word sense disambiguation in lexical chaining. In *Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCAI–03)*, pages 1486–1488, Acapulco, Mexico.

Galley, Michel and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 180–187, Rochester, NY.

Grefenstette, Gregory. 1998. Producing Intelligent Telegraphic Text Reduction to Provide an Audio Scanning Service for the Blind. In *Proceedings of the AAAI Symposium on Intelligent Text Summarization*, pages 111–117, Stanford, CA.

Grosz, Barbara J., Scott Weinstein, and Aravind K. Joshi. 1995. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

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

Hirst, Graeme and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Database*. MIT Press, Cambridge, MA, pages 305–332.

Hori, Chiori and Sadaoki Furui. 2004. Speech summarization: An approach through word extraction and a method for evaluation. *IEICE Transactions on Information and Systems*, E87-D(1):15–25, 1.

Jing, Hongyan. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the 6th conference on Applied Natural Language Processing*, pages 310–315, Seattle, WA.

Kibble, Rodger and Richard Power. 2004. Optimising referential coherence in text generation. *Computational Linguistics*, 30(4):401–416.

Knight, Kevin and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107.

Kupiec, Julian, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR-95*, pages 68–73, Seattle, WA.

Lin, Chin-Yew. 2003. Improving summarization performance by sentence compression—A pilot study. In *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages*, pages 1–8, Sapporo.

Lin, Dekang. 2001. LaTaT: Language and text analysis tools. In *Proceedings of the first Human Language Technology Conference*, pages 222–227, San Francisco, CA.

Mani, Inderjeet. 2001. *Automatic Summarization*. John Benjamins, Amsterdam.

Mani, Inderjeet, Thérèse Firmin, David House, Gary Klein, Beth Sundheim, and Lynette Hirschman. 2002. The TIPSTER SUMMAC Text Summarization Evaluation. *Natural Language Engineering*, 8:43–68.

Mani, Inderjeet, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 558–565, College Park, MD.

Mann, William C. and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Marciniak, Tomasz and Michael Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL–2005)*, pages 136–143, Ann Arbor, MI.

Marcu, Daniel. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, Cambridge, MA.

Martins, André and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 1–9, Boulder, CO.

Martins, André, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 342–350, Suntec.

McDonald, Ryan. 2006. Discriminative sentence compression with soft syntactic constraints. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 297–304, Trento.

Miltsakaki, Eleni and Karen Kukich. 2000. The role of centering theory's rough-shift in the teaching and evaluation of writing skills. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 408–415, Hong Kong.

Morris, A., G. Kasper, and D. Adams. 1992. The effects and limitations of automated text condensing on reading comprehension performance. *Information Systems Research*, 3(1):17–35.

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

Nguyen, Minh Le, Akira Shimazu, Susumu Horiguchi, Tu Bao Ho, and Masaru Fukushi. 2004. Probabilistic sentence reduction using support vector machines. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 743–749, Geneva.

Olivers, S. H. and W. B. Dolan. 1999. Less is more; eliminating index terms from subordinate clauses. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 349–356, College Park, MD.

Ono, Kenji, Kazuo Sumita, and Seiji Miike. 1994. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 344–348, Kyoto.

Orăsan, Constantin. 2003. An evolutionary approach for improving the quality of automatic summaries. In *ACL Workshop on Multilingual Summarization and Question Answering*, pages 37–45, Sapporo, Japan.

Poesio, Massimo, Rosemary Stevenson, Barbara Di Eugenio, and Janet Hitzeman. 2004. Centering: a parametric theory and its instantiations. *Computational Linguistics*, 30(3):309–363.

Press, William H., Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1992. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK.

Punyakanok, Vasin, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1346–1352, Geneva.

Riedel, Sebastian and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 129–137, Sydney.

Riezler, Stefan, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of*

the Association for Computational Linguistics, pages 118–125, Edmonton.

Roark, Brian. 2001. Probabilistic top–down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

Roth, Dan and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the 8th Conference on Computational Natural Language Learning*, pages 1–8, Boston, MA.

Scott, Donia and Clarisse Sieckenius de Souza. 1990. Getting the message across in RST-based text generation. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*. Academic Press, New York, pages 47–73.

Sjorochod'ko, E. F. 1972. Adaptive method for automatic abstracting and indexing. In *Information Processing 71: Proceedings of the IFIP Congress 71*, pages 1179–1182, Amsterdam.

Tetreault, Joel R. 2001. A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics*, 27(4):507–520.

Teufel, Simone and Marc Moens. 2002. Summarizing scientific articles—Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–446.

Turner, Jenine and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 290–297, Ann Arbor, MI.

Vanderbei, Robert J. 2001. *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, Boston, 2nd edition.

Walker, Marilyn, Aravind Joshi, and Ellen Prince. 1998. Centering in naturally occurring discourse: An overview. In *Centering Theory in Discourse*. Oxford University Press, Oxford, pages 1–28.

Winston, Wayne L. and Munirpallam Venkataramanan. 2003. *Introduction to Mathematical Programming*. Brooks/Cole, Independence, KY.

Wolf, Florian and Edward Gibson. 2004. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 383–390, Barcelona.

Zajic, David, Bonnie J. Dorr, Jimmy J. Lin, and Richard M. Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*, 43(6):1549–1570.