Saarland University

Department of Computational Linguistics

Seminar: **Recent Developments in Computational Discourse Processing**

Text Compression & Text Simplification

How discourse information can improve both tasks

Author:
Patrick Carroll
Matriculation: 2548790

Supervisors:
Dr. Alexis Palmer
Annemarie Friedrich

October 2, 2014

Abstract

The abstract will probably have to be filled out at the very end of writing the paper, because I am not sure what shape it will take until I have some things written down on the page. Hopefully this is not an issue.

The goal of this paper is to provide an overview of two related NLP tasks: Text reduction and text simplification. Both tasks share a common goal of preserving important information, while reducing the complexity of the text as a whole. This reduction techniques that are currently being developed in the Natural Language Processing world.

Contents

1 Introduction

In the Natural Language Processing community there is currently an open frontier of research focused on different ways in which a text can be condensed from it's original versions into some form of compressed/reduced/summarized output. The general goal of these tasks is to process a text in such a way that important information is preserved but some aspect of complexity is reduced. Over the course of this paper, I intend to present and compare two closely related tasks in this field: text compression and text simplification. While exploring these tasks, I also intend to focus on the use of discourse level information which, when applicable, represents a promising new avenue of research for both text compression and text simplification tasks.

Let me begin by going over some basic definitions of what each task entails. I'll start with text compression, which according to Clarke and Lapata (2010) encompasses automatic methods for shortening sentences with minimal information loss while preserving their grammaticality. It is important to point out that this definition refers to the process of shortening a text's length on a word by word basis, rather than a sentence by sentence basis, (as is the case with most text summarization systems). In this paper I will present three publications as examples of text compression, The first two papers from Knight and Marcu (2000) and Galley and McKeown (2007) describe noisy channel based approaches which performs compression on isolated sentences without taking surrounding context into account. The third system is from Clarke and Lapata (2010) and performs compression across a whole document using discourse information.

For text simplification, the task can be defined as taking a sentence as input and aiming to produce a simplified version with a less complex vocabulary and/or sentence structure while preserving the main ideas from the original text (Feng, 2008). In contrast to text compression, the procedure for simplification is not as clear cut as just removing words (although deletion may play some role). Often the task calls for splitting long sentences into a series of smaller sentences, replacing semantically difficult or ambiguous words with simpler ones, or re-phrasing a sentence to change it's syntactic structure (Coster and Kauchak, 2011). For this paper I will focus on 2 examples. The first example from from Siddharthan (2006) uses discourse information to break down syntactically complex sentences into smaller sub-sentences. The second example from Coster and Kauchak (2011) attempts to reduce the semantic complexity of a text, and frames the simplification task as a translation from standard English to simple English using parallel corpora taken from Wikipedia.[1]

Continuing on from these basic definitions, the following sub-sections of the introduction will discuss some of the the real world applications motivating these tasks, the models

[1] http://simple.wikipedia.org, http://en.wikipedia.org

which are used to describe the problem of each task, and the data used for training and testing. Following the introduction, section 2 will describe in greater detail how each of the systems work, and how they relate to one-another. Section 3 is reserved for general discussion on the use of discourse information in the tasks, conclusion on the current state of these technologies, and speculation on future work.

1.1 Applications

As mentioned in the previous section, both tasks share an over-arching purpose of transforming a text from it's original content into a condensed output. How that condensed output can be put to use is the question being explored in this sub-section. In many cases, the applications for text compression have a certain amount of overlap with text simplification an vice-versa. One can view this as indicative of how closely the tasks are related and how they lie on a spectrum of different ways in which text complexity is reduced.

Of the two tasks, compression has the most broad set of applications. Over the course of it's development text compression has been proposed as a post processing step for improving text summarization systems (Jing, 2000; Knight and Marcu, 2002) , as a means of reducing text length for PDA's (Corston-Oliver, 2001), as a component of subtitle generation (**?**), and as a reading aid for the blind (Grefenstette, 1998). From this diverse range of applications, one can see that compression is used not only to improve information density problems like summarization, but can also be put to use increasing readability when text length is limited. One of the reasons compression finds so many useful outlets is that the task's purpose of deleting superfluous words is a relatively universal tool which can be very easily adapted to many different domains. Because the task of simplification involves several different edit operations, it appears to have more narrowly defined range of applications at present.

While transforming texts for greater readability can be achieved by text compression (as mentioned earlier for subtitles and PDAs), text simplification is a tool specifically envisioned for such situations. Where it differs from compression is that by reducing semantic or syntatic complexity, rather than just length, it can be a more targeted approach to increasing readability for specific groups of people. For instance it has been proposed as a tool to help people with reading comprehension difficulties such as children and foreign language learners, or those with cognitive impairments such as aphasics(Feng, 2008). In such applications, psychological and pedagogic theories are taken into account in order to try and reduce memory load and cognitive processing for readers. Text simplification has also generated interest in the field of information extraction as a means to improve document recall for some specialized document collections such as medical publication. In theses cases, simplification is used to break up large sentences with the idea being

that search and information extraction tasks show better results on smaller sentences. (Jonnalagadda and Gonzalez, 2010).

1.2 Models

This subsection is intended to describe how the problem of each task is modeled. Beginning with compression, the task is typically viewed as a word deletion operation. According to Knight and Marcu (2002), given an input sentence of words $x = x_1, x_2...x_n$ the aim is to produce a compression which is some sub-set of these words. The rules for how that sub-set of words is reached will differ for each specific system discussed. In the first two papers on compression (Galley and McKeown, 2007; Knight and Marcu, 2000) the rules model a translation from a compressed sentence to a long sentence. In the third paper (Clarke and Lapata, 2010), the rules for compression are modeling a scoring function for candidate compressed texts with a set of constraints which penalize ungrammaticality, and encourage retention of discourse relevant words.

The general model for the problem of text simplification expands upon the deletion problem mentioned in the previous paragraph. In addition to the deletion operation, the operations of word substitution, reordering, insertion, and sentence splitting may also be included in modeling the task(Siddharthan, 2006; Coster and Kauchak, 2011). Most simplification system only use *some* of the operations mentioned, so it's important to point out that it's difficult to compare different simplification system models developed for different purposes. In the first simplification paper from Siddharthan (2006) the authors use hand coded rules based on syntactic and discourse information to determine how long sentences should be split into smaller sub-sentences. In a second step they address issues of cohesion for the sub-sentences by inserting and re-ordering words. The second simplification system from Coster and Kauchak (2011) is more focused on semantic simplification, thus it models the transformation from standard to simplified text as a phrase based translation, somewhat similar to the translation model in the compression papers.

1.3 Data

In each of the papers presented, there must be some data upon which the models are trained and/or tested. This section is

Highlight the kind of data used to train and/or model the problem for each text reduction task. Is the data directly used to train a system, or is it simply used as a frame of reference for un-supervised learning. What kind of data is used for validation and evaluating the systems?

2 Compression Related Tasks

2.1 Text Compression as Translation

For the NLP tasks of text compression and text simplification, an expedient way to make progress towards a working system has been to adapt existing methods from other NLP tasks. In the case of text compression, the noisy channel model has been a promising method used by several different researchers because of it's suitability to tasks which require a conversion from one string to another (Knight and Marcu, 2000; Galley and McKeown, 2007). In this sub-section we will discuss the theoretical underpinnings of a Noisy Channel model as proposed by Knight and Marcu (2000) and then shift our attention to some improvements by Galley and McKeown (2007) which builds off of Knight and Marcau's system by adding variable levels of lexical and phrase based information. This sub-section is intended to give the reader a close look at some data driven approaches to text compression which operate only on the sentence. From this starting point we can later explore how the noisy channel model relates to some text simplification systems, and how it diverges from discourse based systems in both compression and simplification.

Knight and Marcu (2000) were some of the first researchers to propose the noisy channel model as a solution to the compression problem. In their paper they provide a thorough and illustrative description of how the noisy channel works in general, which serves as the conceptual grounding for their system, as well as the more current system by Galley and McKeown (2007). In the paper, the authors assume that a short string (the ideal compression) is transmitted over a channel and at the other end a long string (the uncompressed message) is received. Because the channel for transmitting the message is noisy, the original short message is corrupted and ends up being transformed into the long one which can be observed. The goal is to reason backwards from the long string as to what is the most likely short string which created it (see figure 1). In other words if c is a compression in the set of all grammatical compressions C, and f is the full text observed, then one is trying to find $\hat{c} = argmax_{c \in C}\{p(c|f)\}$. This equation can be further broken down into a **translation model** and a **language model** as in the case ofKnight and Marcu (2000), or in the case of Galley and McKeown (2007) the two models are collapsed into one single translation model for purposes of easier computation within their framework.

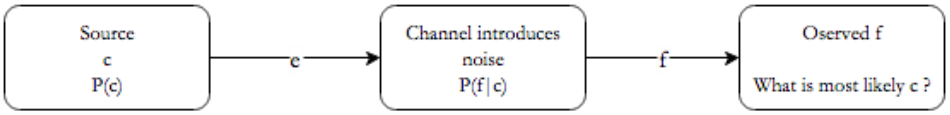

Figure 1. The noisy channel pipeline.

In any implementation of the noisy channel model, the translation model is the key to describing the probability that one string is transformed into another across the channel. In a classic example of the noisy channel model for machine translation, obtaining these probabilities is a well established task of collecting counts of string to string translations from a corpus of aligned sentence pairs. However, because word deletion can be difficult to accurately model in a string to string translation, this approach was deemed inadequate to represent sentence compression. In order to solve this problem, Knight and Marcu (2000) instead use a syntax based approach which models a translation between sub-trees described by a **Synchronous Context Free Grammar** (SCFG). SCFGs can be defined as context free grammars whose productions have two right hand side rules, one representing the source constituents, and the other representing the target constituents, which in the case of text compression will be some sub sequence of the source side (Galley and McKeown, 2007). The following is an example of a deletion rule in a SCFG for removing a prepositional phrase:.

$$\text{VP} \rightarrow \langle \text{ VBD PP PP, VBD PP } \rangle$$

Figure 2. An example rule from an SCFG

In order to train a SCFG language model, parallel sentences must be obtained from a corpus to estimate the likelihoods for deletion rules in the grammar. Both papers used the Ziff-Davis corpus which is a collection of technical documents paired with abstracts. Knight and Marcu (2000) only use a small fraction of the corpus (1.75% of abstract sentences), because they follow the strict definition of compression as a series of word deletions, and this means they can only include sentence pairs where the compressed sentence is an exact subset of words from the source sentence. Furthermore, of the small fraction of the corpus being used, they can only train rules for the grammar when there is a clear alignment between sub-trees in the source and compression sentences. This leaves them with a rather small set of data to train their system on. Galley and McKeown (2007) decide to loosen their criteria of sentence compression and allow for a limited number of word substitutions edits in addition to word deletions. This provided a much larger set of training data (up to 25% of summary sentences when allowing 6 substitutions per sentence). It's interesting to note that if even more edit rules were allowed in training the system, such as insertion, or re-ordering, the resulting SCFG rules would be moving away from a model of pure compression towards something more like text simplification. This shows that sentence compression could be thought of as a special case of simplification.

By using SCFGs, the papers make some implicit assumptions about what kind of information will play a role in deleting words from a source sentence. For instance, the focus on deletion of constituents rather than words makes the system blind to any lexical information which may bias towards retaining or deleting a constituent from a sentence. To

improve upon this Galley and McKeown (2007) decided to expand the annotation in their SCFGs by tagging the lexical head and POS of a syntatic category. By adding lexical head information the system can more accurately predict for instance, that some PP's are compliments of a verb (and likely to be preserved) while other PP's are adjuncts and better candidates for deletion (see figure 3).

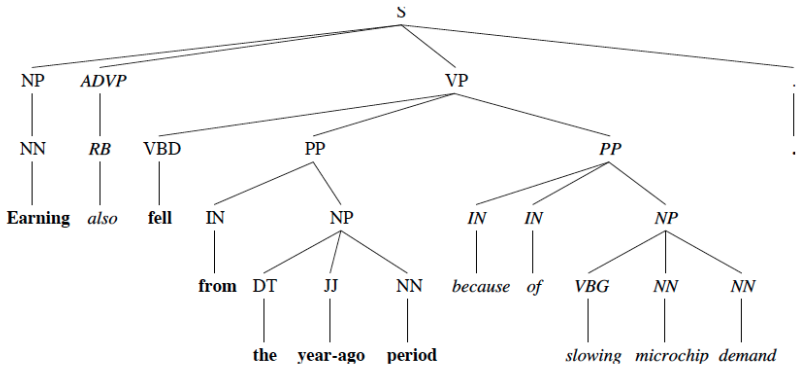

Figure 3. a penn treebank example of PP compliment and PP adjunct.

The authors are also interested in using embedding depth to predict the importance of a constituent. In order to model this, they add annotation to label the parents of constituents to varying depth, for example an NP beneath a PP, VP, and S parent node would be labeled NPˆPPˆVPˆS with an embedding depth of 3. They assume that constituents embedded deep within a PP, for example, would be good candidates for removal based on their position and depth in the sentence. This idea of phrase embedding depth also plays an important role in the next system being discussed, although used in a slightly different manner. While these extra features were found to improve the accuracy of the Galley and McKeown (2007) model above Knight and Marcu (2000), it is important to mention that both systems are only compressing individual sentences, and base their deletion decision almost exclusively on syntactic information. As one can imagine, there are many situations where text compression may benefit from more information than just parse trees. Concepts like word frequency scores, lexically relations, and sentence 'centers' may all contribute to a more nuanced and cohesive compression that what has been presently discussed. These concepts and more will be explored in the next sub-section, which will cover a discourse driven approach to text compression.

2.2 Text compression With Discourse Constraints

The conceptual framework for the compression system used by Clarke and Lapata (2010) diverges significantly from the noisy channel systems in the previous sub-section. the

authors re-imagines the task as an optimization problem: Given a string of text, retain the words which maximize a scoring function. The scoring function is bound by a series of competing constraints, including rules about enforcing grammaticality, keeping text to a certain length, and retaining informative words based on sentence and discourse level information(Clarke and Lapata, 2010). In order to solve for the optimal score, the scoring function and constraints are represented as linear inequalities and are evaluated by **Integer Linear Programming** (ILP). ILP can be thought of as a type of general problem solver tool which efficiently searches for an optimal integer solution to linear functions. This approach frees up the system to perform more flexible compression than was possible in the noisy channel approach. This allows for more variable compression rates, compressions of text of arbitrary length (rather than just sentences), and the ability to model any important information for compression, such as discourse information, as a constraint to the scoring function.

The first part of the scoring function is a significance score aimed to boost the value of topic related words (assumed to be nouns and verbs). The significance score attempts to get the gist of how important a word is the context of the document, and the sentence. The first part of the significance equation is something similar to **tf • idf**, and compares the word's document frequency to it's frequency in a training corpus of similar documents and determines it's relative importance. The second part of the equation gives more importance to words depending on the depth of **embedding** of the clause they are found in. The intuition from the authors is that a content word in a deeply embedded clause carries more semantic content, and thus should have higher chances of retention. This representation of an embedding score is quite different from the example by Galley and McKeown (2007) who measured constituent depth rather than clause depth. But it shows that in both cases, looking at where a word lies in the 'vertical' space of a parse can be helpful to determine it's importance to the final compression.

Next, the backbone of the scoring function is a language model of candidate compressions which operates in tandem with a set of constraints. The language model calculates a score for all candidate compressions, while the constraints dictate which compressions are vailid, and penalize/reward certain compressions based on syntax and discourse content (Clarke and Lapata, 2010). The first set of constraints on the language model are to ensure that any candidate compression is a well formed in the most basic technical sense, meaning that the sentence has a start and end point, and than only one word occupies each position in the sentence. The next set of constraints are sentence level syntactic constraints developed by the authors in a previous paper (Clarke and Lapata, 2008). These constraints are hand coded rules intended to preserve the meaning and structure of the original sentence as much as possible. The rules in general describe situations where if a certain element is included (negations, adjectives, adverbs, determiners, prepositions) than the head of that element must also be included so it's not left floating in a sentence

without context. As an example, if an argument is present in the compression, than the verb to which that argument belongs must also be included. This bottom up approach to retaining constituents contrasts with the previous papers which had a top down deletion rule as illustrated by figures (3) (2). The choice to use hand coded syntax rules rather than a data driven approach means the system may miss some useful syntax rules derived from the data, but is more insulated from "noisy" rules which are observed very infrequently.

The third set of constraints are related to discourse information and are used to prioritize retention of words which support coherence within the document. Coherence is an important concept in discourse which relates to the overall intelligibility of the message in a text. Because global coherence is quite difficult to model automatically (attempts have been made using Rhetorical Structure Theory ***CITE), the authors instead use two complimentary theories of discourse which attempt to model local cohesion as a proxy for global coherence. The first theory is **lexical chains**, which can be thought of as groups of semantically related words from the document which represent lexical cohesion ***Cite (Morris and Hirst 1991). Main topics for the paper are identified as long lexical chains with strong semantic relation among them (see figure 4). The next discourse theory used is **centering theory**. This theory operates on the idea that within a section of text, typically a sentence, there exists an entity which is more salient than the others and considered the focus of the sentence. The center is identified according to it's grammatical function, and it's relation to entities in the preceding and following sentences (see figure 4). In using these discourse constraints to retain words which show high cohesion, the system in effect also gains a secondary grammar check which strongly rewards retention of the subject and objects in a sentence.

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 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 Etnas slopes.

Figure 4. lexical chains are listed in italics with sub-script and centers are written in bold

The use of discourse information in many NLP tasks at present still faces a challenge of reliably and accurately identifying the elements relevant to discourse. This paper provides a good example of applying discourse information in a manageable scale, such as searching for local cohesion, rather than attempting to define the global discourse structure of the whole document. The system is also designed in such a way that training data comes from a single corpus, rather than needing aligned pairs of source and compression sentences. This design is much more insulated against issues of sparse data as seen in the previous examples from Galley and McKeown (2007) and Knight and Marcu (2000). This is an

important consideration for any compression or simplification system, considering that quality results may be more limited by available data rather than any flaw in the design of the model itself.

2.3 Simplification with Discourse Constraints

Moving now from text compression to text simplification, the first example being presented is from the a paper by Siddharthan (2006). The system described in this paper is designed to reduce the syntactic complexity of a document while also trying to retain text cohesion. The author motivates the use of discourse level information for the task by pointing out several problems which can arise from compression or simplification without discourse information. The first concern is that a loss of cohesion could make the text more difficult to comprehend, which in effect, is the opposite of what simplification systems aim to achieve. A system blind to discourse information also runs the risk of changing the intended meaning. This concern is echoed in the previous system which also tries to achieve better coherence in the compression by use of lexical chains and centering theory.

The scope of simplification as presented in this paper is to divide long sentences into smaller sentences, and then to resolve issues of cohesion through sentence (re)ordering and addressing pronominal links. The architecture of the system is divided into 3 steps: **analysis, transformation,** and **regeneration** (see figure for an illustration of the architecture). In the analysis stage, markup is performed on the text for sentences boundaries, POS tags, noun chunks, pronoun resolution, and boundaries and attachment for clauses/phrases. This step is pre-processing so that the transformation stage can apply rules for breaking up a sentence into sub-sentences. Each time a sentence is split, it's sent to the regeneration stage to address issues of conjunctive cohesion such as sentence order, referring expressions, and adding cue words. The transformation stage is repeated recursively, meaning that each sub-sentences is placed back on the stack of sentences to split until no further splitting rules apply. All minimally reduced sentences are passed one final time to a different part of the regeneration stage, at which point anaphoric cohesion is resolved.

The rules for how sentences are split in the transformation phase are hand written and are similar to a Context Free Grammar. However this rule set differs from the previous SCFG seen in Knight and Marcu (2000) in a few ways. To illustrate, the rule displayed in figure 5 shows a sentence comprised of parts $VWXYZ$ where W is a noun phrase and Y is a relative clause attached to the noun in W. The rules states that the sentence can be split into sub sentences (i) and (ii), and are linked by the relation $RELPR$. Each time a sentence is split, the resulting form is a triplet (a, R, b) which shows one reduced sentence a as a nucleus connected to the other reduced sentence b by relation R. The

creation of the rule set with relations and the use of nucleus/satellite terminology is based on **Rhetorical Structure Theory** (RST), which is a comprehensive discourse theory used to describe how sentences relate to one another across a document. It's important to note that the author does not attempt to parse the source text into RST structure for simplification, but instead chooses a more tractable task of matching phrases in a sentence to hand picked rhetorical relations.

Once sentences have been split by a rule in the transformation stage, they must be properly ordered to maintain coherence. For the sake of simplicity, re-ordering occurs each time a sentence is split, rather than

$$\langle s\rangle VW^{n}_{NP}[_{RC}RELPR^{\#n}Y]Z.\langle s\rangle \rightarrow \begin{matrix}(i)\langle s\rangle VWXZ.\langle /s\rangle \\ (ii)\langle s\rangle WY.\langle /s\rangle\end{matrix}$$

Figure 5. a simplification rule for the system by Siddharthan.

3 Discussion

3.1 Conclusion

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

3.2 Thoughts on Future Work

References

Clarke, J. and Lapata, M. (2008). Global inference for sentence compression an integer linear programming approach. *J. Artif. Int. Res.*, 31(1):399–429.

Clarke, J. and Lapata, M. (2010). Discourse constraints for document compression. *Comput. Linguist.*, 36(3):411–441.

Corston-Oliver, S. (2001). Text compaction for display on very small screens. In *Proceedings of the NAACL Workshop on Automatic Summarization*, pages 89–98. Association for Computational Linguistics.

Coster, W. and Kauchak, D. (2011). Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Feng, L. (2008). Text simplication: A survey. Technical report, CUNY.

Galley, M. and McKeown, K. (2007). Lexicalized markov grammars for sentence compression. In *HLT-NAACL*, pages 180–187.

Grefenstette, G. (1998). Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working notes of the AAAI Spring Symposium on Intelligent Text summarization*, pages 111–118.

Jing, H. (2000). Sentence reduction for automatic text summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 310–315, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jonnalagadda, S. and Gonzalez, G. (2010). Biosimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction. In *AMIA Annual Symposium Proceedings*, volume 2010, page 351. American Medical Informatics Association.

Knight, K. and Marcu, D. (2000). Statistics-based summarization-step one: Sentence compression. In *AAAI/IAAI*, pages 703–710.

Knight, K. and Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107.

Siddharthan, A. (2006). Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.