Saarland University

Department of Computational Linguistics

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

Compression, Simplification, Fusion:

An Overview of Text Compression Related 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 three major text 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 texts can be condensed from their original versions into some form of compressed/reduced/summarized output. The general goal of these tasks is to process a text or texts 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 give an overview of three of these tasks: text compression, text simplification, and sentence fusion. While exploring these tasks, I also intend to focus on the use of discourse level information, which, when applicable, represents a promising means of identifying important information in these related 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 two publications as examples of text compression, The first is an earlier system from Galley and McKeown (2007) which performs compression on isolated sentences without taking surrounding context into account. The second system is from Clarke and Lapata (2010) and performs compression across a whole document using discourse information.

Moving next to 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 (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]

Next on to the task of sentence fusion. Sentence fusion can be defined as creating a summary of a multi-document collection by fusing parts of related sentences together (Filippova and Strube, 2008). The process for doing so is best divided into two steps.

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

In the first step, often referred to as paraphrase detection, the aim is to search a multi-document database and identify clusters of equivalent sentences or sentences fragments (Regneri and Wang, 2012). In the second step, commonly referred to as the fusion step, the aim is to create a summary sentence for each paraphrase cluster which maximizes common information across the sentences, while minimizing redundancy. This is done by combining parts of sentences from a cluster and inserting other function words to ensure a grammatical sentence upon output.(Filippova and Strube, 2008). In this paper I will cover an end to end approach of the task by Filippova and Strube (2008), and a stand alone paraphrase detection system by Regneri and Wang (2012) which makes use of discourse information for better paraphrase clustering.

Continuing on from these basic definitions, the remaining sub-sections of the introduction will look in greater detail at the real world applications of these tasks, the models 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 mentioned above work. This section will also contrast the differences between those system which use discourse information, and those which do not. Section 3 is reserved for discussion of the relations between the tasks, conclusion on the current state of these technologies, and speculation on future work.

1.1 Applications

As mentioned in the previous section, all three 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 I intend to explore in this sub-section. As one will see, many of the tasks have some overlap in their applications, or they share common goals but vary in their approaches to solving them. This is indicative of how the tasks are related and lie on a spectrum of different manners in which a text can be reduced.

Of the three tasks, compression has the most broad set of applications, so it serves as a good starting point. 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) , a means of reducing text length for PDA's (Corston-Oliver, 2001), 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 can be very easily adapted to many different domains. The other two tasks involve

more complex operations than just word deletion, and as a result, have a more narrowly defined range of applications at present.

While text compression was envisioned as a general tool to condense any sort of text summary, sentence fusion was developed with a more narrow scope of summarizing multi-document collections. It's need arose from the problem that extractive summarization systems often have a trade off between producing long winded summaries which adequately cover important information, or summaries which sacrifice information in favor of brevity (Filippova and Strube, 2008). Of the three task being discussed, sentence fusion is by far the most specifically focused on the single application of summarization, however, many of the individual components developed for sentence fusion can find use in other application. For instance one technique for collapsing sentences together, dependency graph pruning, has also been found to be useful for sentence compression as well (**?**). And the clustering of sentence paraphrases can be applied to a wide variety of applications such a recognizing textual entailment (**?**) and natural language generation(**?**).

Transforming texts for greater readability is the main has also been a popular topic of research, and has made good use of text compression and text simplification. In the case of text simplification, the goal of improving readability is often focused on 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, the purpose of text simplification is to reduce the semantic complexity of a text, and/or to alleviate some memory load issues by breaking complex sentences down into shorter, more easily processed sentences. Some other readability applications include using text compression to reducing text length for display on PDA's (**?**) or as a reading aid for the blind (**?**). In these cases, compression will suffice because it's readability issues arise from text length rather than text difficulty.

1.2 Models

This subsection is intended to describe how the problem of each task is modeled. By looking at what operations are to be performed, and the constraints governing those operations, one can see how the three tasks are related.

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 constraints in this model can be viewed conceptually as the rules which decide which words are kept, and which are deleted.

The approach used to solve the problem of text simplification expands upon the deletion

problem mentioned in the previous paragraph. In addition to the deletion operation, the operations of reduction, rewording, reordering, and insertion may also be included in modeling the task(Siddharthan, 2006)(Coster and Kauchak, 2011). In terms of constraints which determine what operation to perform, the example system from Siddharthan (2006) uses hand coded rules based on syntatic information, while the system from Coster and Kauchak (2011) is a data driven approach which learns rules for the operations from parallel corpora or standard and simple texts.

The approaches used for sentence fusion must be viewed as two sub tasks. The task of paraphrase detection uses a

1.3 Data

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?

This section will focus on the data used to train

1.4 Natbib citations

Within a text, you can say that Lin and Pantel (2001) found out something. Or you can just state the thing, and then put the author in parentheses (see Szpektor et al., 2004).

2 Compression Related Tasks

2.1 Text Compression as Translation

For relatively recent NLP tasks such as text compression or simplification, an expedient way to make progress towards a successful implementation is to find pre-existing models which can easily be adapted to a new task. One of the most fruitful models for other NLP tasks has been the noisy channel model. Because the noisy channel model is well suited to tasks which require a conversion from one string to another, like in the case of translation, it has also been seen as a natural candidate for text compression(**?**). The systems for text compression which use this noisy channel model claim a fair amount of success and thus represent one important approach which can be compared against the discourse information based system presented later in this section. The system which I will cover here is from the article Lexicalized Markov Grammars for Sentence Compression by Galley and McKeown (2007). I've selected it because it's a relatively recent example of a Noisy Channel based approach which also covers in some depth the incremental improvements over past Noisy Channel text compression systems.

When attempting to solve any NLP task using a Noisy channel model,it's helpful to first establish what element is playing what role within the model. So to lay the conceptual groundwork, the noisy channel model assumes that a short string (the ideal compression) is transmitted over a channel and at the other end a long string (the un-compressed 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(**?**). This is done by using parallel corpora to create a *translation model* and a *language model*. The *translation model* describes what kinds of transformations from a long string to a short string are permitted, and what is their likelihood. The *language model* is used to determine the likelihood that a candidate compression output from the *translation model* is a well formed sentence in the language.(**?**)

All noisy channel text compression operates on the general principles described in the previous paragraph. However,the manner in which the translation and language models are represented and trained play a large role in the effectiveness of the system. The translation model used in Galley and McKeown (2007) represents the compression of a sentence as

builds on previously successful noisy channel model approaches like that from which model the deletion problem as a kind of translation from a the noisy channel model described in

That's because text compression in some sense can be viewed as a translation from a long

sentence to a shorter sentences while preserving the important information.

? syntax driven approaches such as that used by is a syntax driven approach which represents rules for deletion as grammar rules in a synchronous context free grammar. This approach of builds upon previous The system described in this article performs compression on a sentence by sentence basis without any knowledge of the discourse structure or relation between

In order to arrive at suitable constraints for deletion, several approaches have been proposed. One example from (Knight and Marcu, 2002) models compression as a translation from a verbose sentence to a sparse one. In this approach the noisy channel model is used to find the most likely compression out of a set of many possible compressions. In the paper by Clarke and Lapata (2010) which I will be covering in section 2, 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 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.

2.2 Simplification

2.3 Paraphrasing/Fusion

Figure 1. The saarland uni logo.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus.

Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

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. (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.

Filippova, K. and Strube, M. (2008). Sentence fusion via dependency graph compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 177–185, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

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

Lin, D. and Pantel, P. (2001). DIRT - Discovery of Inference Rules from Text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-01)*, pages 323–328, San Francisco, CA.

Regneri, M. and Wang, R. (2012). Using discourse information for paraphrase extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 916–927, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Szpektor, I., Tanev, H., Dagan, I., and Coppola, B. (2004). Scaling web-based acquisition of entailment relations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 41–48, Barcelona, Spain.