NLP Assignment 4

Anand Kamat

ID# 260773313

Question1

The leading algorithm just extracts the first few sentences for the summary. Although the leading algorithm doesn't quite summarize the articles well enough however in several cases the first sentence of the passage proves to be the most relevant and the leading algorithm always has that. The redundancy does work for the original algorithm. It reduces the probability of the words a lot. The redundancy test does prove to be true as the simplified algorithm had two sentences talking about the traction control of the car. The methods seem quite successful in summarizing the articles as the summaries generated are pretty close to the 'gist' of the articles.

Summarization Evaluations:

•Does it accurately reflect the original content?

The leading may not be the best when it comes to summarizing the articles but original and the simplified do a pretty good job at reflecting the original content. The original algorithm does well preventing redundancies.

•Does it include the most important content?

Again the leading algorithm does not contain *all* the relevant information but a better summary is given by the original sumbasic algorithm as well as the simplified algorithm.

•Does it include non-redundant content?

The leading may include redundant content as the words are not evaluated or weighted however the simplified algorithm may suffer from redundant content as the weights are not updated. The original algorithm doesn't not show significant redundant information as the weights are updated as they are evaluated.

•Grammaticality of the individual sentences

The grammaticality of all the algorithms are fairly correct as they are extractive based algorithm. Their grammaticality would be pretty much the same as the grammaticality of the aritcles.

•Coherence of the output

The leading displays a good level of coherence of the output as it just extracts the first few sentences directly. The original algorithm and the simplified may show varied results here as they are based on the score the sentence gets instead of evaluating how the sentences are correlated.

Selecting the order of the sentences in the summary is very tricky to do. There are many approaches one can take to perform this task. The leading algorithm obviously wouldn't be considered in this as the order would the same as that in the article. In case of sum basic we can choose to have the sentences observed earlier and at the end of the article to be given more importance over the others. These sections (introduction and conclusion) are the passages where the overall content is described very well. The order can be biased with respect to them. Sentences in the summary acquired from the introductory passage can be shown first (by giving more weightage) and the last section sentences can be the last few sentences in the summary. The only challenge this may face would be redundancy between sentences in the introductory section and the conclusion. This could be avoided by the squared probability update function we have seen in the original algorithm.

Question 2

Automatic sentence compression can be broadly described as the task of creating a grammatical summary of a single sentence with minimal information. Most prior work deal with sentence compression using word deletion. The simplification renders the task computationally feasible, allowing efficient decoding using a dynamic program. However, since simplification renders the task somewhat artificial, there are other rewrite operations besides deletion used such as: reorder, substitution and insertion. Hence the paper focuses more on generating abstracts rather than extracts. The two big difficulties of this approach are: finding an appropriate training set and the modelling task itself.

The authors generated a corpus for abstractive compression by having annotators compress sentences while rewriting them. They collected 30 newspaper articles from the British National Corpus (BNC) and the American News Text Corpus. The annotators were guided as to how they can paraphrase the sentences while preserving the most important information and also using the rewriting operations mentioned.

They also present an tree-to-tree transducer capable of transforming an input parse tree into a compressed parse tree. This approach can account for structural mismatches and is trained discriminately. Their work is presented as an extension of the model developed by Cohn and Lapata in 2007. They formulate sentence compression as a tree-to-tree rewriting task. Each grammar rule is assigned a weight and those weights are learnt in discriminative training. The grammar allows us to search for all sister trees for a given tree. The maximization step over the space defined in page 3 is solved using dynamic programming. The model is trained using SVM^{Struct}. The grammar extracted is much bigger and noisier than the grammar for solely deletion based compression. They incorporate an n-gram language model as a feature. For the baseline algorithm they use Cohn and Lapata's model and evaluated by eliciting human judgement. Looking at the results in table 4 their abstractive model seems to be performing better than the extractive models in every category.

Advantages: Synchronous grammar provides expressive power to model consistent syntactic effects (like reordering). It is discriminatively trained which allows incorporation of all types of features and it can be tailored to their task using an appropriate loss function.
Limitations: Using the same grammar extraction method as used by Cohn and Lapata and concatenating it with translated bilingual corpus. This seems like a biased approach and not in sync with their data as they described. They used a very small data set which may give high variance in their results. A larger data set would give a more reliable inference. There is always the limitation of unpaid volunteers annotating and rating the data/outputs which can be rectified by hiring experts.
Topics' relevance to:
The grammar discussed in this paper was a context free grammar to allow better expressiveness than say a FSA. The abstractive approach in this paper deals with a NLG methods as it isn't just deletion of words from the sentences. It also reorders, substitutes and inserts words. Extractive Summarization is used when words/sentences are picked from the passage/sentence and presented by the model which this paper has mentioned and has avoided to take another approach by using an abstractive model.
Questions:
What were the rules the annotators were asked to follow? How did they decide on the particular grammar for their model and how did using translated bilingual data the solution to more expressive grammar? Why didn't they approach this problem using an RNN model?