CMPT 825: Natural Language Processing **Spring 2006**

Lecture 22 — Mar 28, 2006

Lecturer: Anoop Sarkar *Scribe: Javier Thaine*

22.1 Text Summarization

Up to this point in the course we have focused on methods for sentence analysis. We now focus on graphs and more complicated models (not just sequences and trees) using methods that look at more than individual sentences. The sentence order is still important, however.

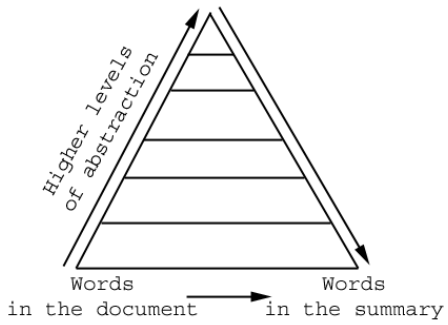
There are many commonalities between the Machine Translation and Text Summarization tasks. Text Summarization is not just one task. One could summarize one document, multiple documents, or even multiple documents in different languages. You could also summarize a single topic within multiple documents.

Here is a broad view of the basic steps of summarization [1]:[1]

- Interpretation of the documents (using named entity tagging, etc.)
- Extraction (using graph-based methods, etc.)
- Condensation (to fit your summary length requirements)
- Presentation in a natural language (called "text generation" or "natural language generation")

We assume that for any particular task, we are summarizing for a particular domain. The processing can be done using some intermediary form instead of natural language (e.g. a logical form), just as in machine translation. We refer to the pyramid we used at the beginning of the term:

[1] Regina Barzilay's slides [1] are referred to frequently in this lecture. She is an expert in summarization and her thesis [2] is useful as a starting point to learn about it.



One can transfer directly from the words in the document to the words in the summary, or one could go one or more levels higher and work at the level of chunks, phrases, or trees. The same criticism, advantages and disadvantages regarding this model which we studied in the first lectures apply here. In practice, most people work roughly at the bottom of the pyramid (everything is done in language). In some examples, however, the input might not be in language and one would want to present it in language (text generation).

As with MT, there are many uses for summarization. One can generate an *indicative* summary. This is useful for someone with many documents to choose from and limited time in which to read them. The user would like to know which to read, and an indicative summary of each with some general notion of the contents would be sufficient, i.e. a complete or coherent summary is not required. The user just wants to know if the document is interesting. The same happens in MT - sometimes MT is just used for IR, in which case the translation need not aim for perfection. Mistranslate one word and chances are that in the same document you will translate it again correctly, thus not affecting your retrieval performance.

On the other hand, an *informative* summary gives you all the salient information in the text (not always well-defined). It tells you, for example, that a news article is about a certain event, and describes it in enough detail that the original document need not be read.

There is another way to divide summarization. One can have an *extractive* summary or an *abstract*. For example, here is an extractive summary followed by an abstract of the Gettysburg Address [1]:

Four score seven years ago our fathers brought forth upon this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal. Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. The brave men, living and dead, who struggled here, have consecrated it far above our poor power to add or detract.

The speech by Abraham Lincoln commemorates soldiers who laid down their lives in the Battle of Gettysburg. It reminds the troops that it is the future of freedom in America that they are fighting for.

In the former, sentences are picked which give the general flow of the document. The latter is a concise summary of the ideas using wording not present in the original document. An abstract is what one would generally want, but an extractive summary is what present technology permits.

22.2 Condensation

Condensation is like leaving the text out in the sun: whatever evaporates is not useful. Headline generation is one possible outcome of condensation. Outlines are similar but more complicated to produce. Condensation is somewhat simpler than summarization. Some applications of condensation are generation of:

- Headlines
- Outlines
- Meeting minutes
- Obituaries (from biographies)
- Movie summaries
- Comparative movie summaries (e.g. description of a general consensus among critics)

22.3 Sentence compression

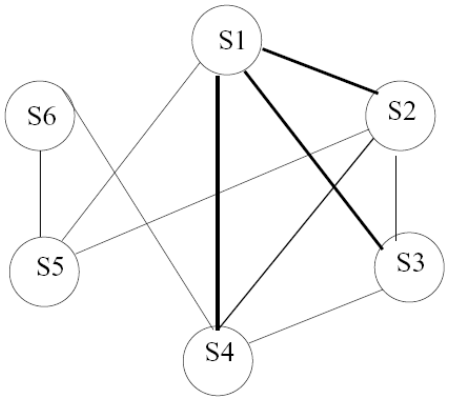
One can think of sentence compression as a machine translation model which is good at deleting things. One could even compress multiple sentences into one. We can use a noisy channel model to think of it: you had a short sentence and it got "corrupted" to the larger version with useless words. One can treat it with a standard Bayesian model.

22.4 Summarization as sentence extraction

The idea is to look at each sentence and decide whether it should be in the summary or not. One could have many features to help decide this: named entities, sentence length, etc. The idea is to learn from a large set of training data in the form of extracted summaries where individual sentences are marked as relevant or not. The training depends highly on the quality of the annotated features in the data. Some specific feature instances that can be used are [1] (an easy to implement algorithm to combine sentential features is given in [6]):

Sentence Length Cut-off Feature	true if sentence > 5 words
Fixed-Phrase Feature indicator	true if sentence contains phrases: *this letter*, *in conclusion*
Paragraph Feature	initial, final, medial
Thematic Word Feature	true if sentence contains frequent words
UppercaseWord Feature	true if sentence contains proper names: *the American Society for Testing and Materials*

One idea is to treat the text as a graph of individual sentences (image taken from [1]):



One can use any standard IR similarity measure on the edges of this graph, i.e. any vector distance (Euclidean, etc.). In practice cosine distance is usually chosen. One is looking for sentences with many incoming edges and a high weight to those edges. Such sentences are very "similar" to the rest and are thus quite representative of the contents of the text and belong in the summary. One needs a way of finding out which nodes in the graph satisfy this. A PageRank algorithm (similar to what Google used in its early days) is good for this (here we should call it SentenceRank). It maintains a distribution based on what's *in* each node and what *points* to it. It is a recursive measure which depends on the SentenceRank of neighboring nodes. One solves the recursion by representing the graph as a matrix, with the rows and columns corresponding to each of the sentences (nodes). According to Radev's work [3], this seems to work better than the obvious ways, such as looking at a node with the highest indegree, etc. As for features, one could use them to obtain an independent ranking for each sentence and multiply this ranking by its SentenceRank to obtain a new measure of its relevance to the summary.

Generally, the more you decrease the size of the summary, the worse it is.

22.5 Information Reordering

The task is to determine the ordering of the sentences in a summary. Here is an example application of reordering [1]. In the second sequence of sentences, the first two provide a good overview of the event. Intuitively it is better

to place them first for a reader to become acquainted with the event. The ordering depends on the substructure contained in the document sentences (e.g. an event will have a certain sequence of subevents).

(a) During a third practice forced landing, with the landing gear extended, the CFI took over the controls. (b) The certified flight instructor (CFI) and the private pilot, her husband, had flown a previous flight that day and practiced maneuvers at altitude. (c) The private pilot performed two practice power off landings from the downwind to runway 18. (d) When the airplane developed a high sink rate during the turn to final, the CFI realized that the airplane was low and slow. (e) After a refueling stop, they departed for another training flight.
(b) The certified flight instructor (CFI) and the private pilot, her husband, had flown a previous flight that day and practiced maneuvers at altitude. (e) After a refueling stop, they departed for another training flight. (c) The private pilot performed two practice power off landings from the downwind to runway 18. (a) During a third practice forced landing, with the landing gear extended, the CFI took over the controls. (d) When the airplane developed a high sink rate during the turn to final, the CFI realized that the airplane was low and slow.

22.6 Single document summarization task

This is largely an information reordering problem. First some sentences are picked out by an extractor. The presenter then has to order them in a coherent way. First the event is described, then its details. A common way to start is to make the first sentence in the summary the same as the first sentence in the original document, which is often a good summary of the event. The presentation style depends highly on the training data. For example, we have training data that shows how earthquake reports are presented.

We think of sentence ordering as a Hidden Markov Model of a special

kind called a *content model*,[4] one in which each *state* represents a topic (it is truly hidden since we don't know the topic). The *emissions* are groups of sentences in the tagging (i.e. the emission generates sentences relevant to the topic). Individual sentences are unordered, but the topics are ordered. The HMM is not fully connected, since one does not transition from any topic to any other. Instead, there is a particular order in which topics can be switched, which is governed by the state transition probabilities. Sentences are clustered using a cosine similarity measure. The clustering is independent of the flow (i.e. it doesn't capture the flow of the document). Some clusters are large (real topics), and others are small (considered noise). "We can use the forward algorithm to efficiently compute the generation probability assigned to a document by a content model and the Viterbi algorithm to quickly find the most likely content model state sequence to have generated a given document; see Rabiner (1989) [5] for details." In essence you give the HMM a set of unordered sentences and it uses Viterbi to find the best ordering. [4] The training data comes from newswire articles, and for testing one must come up with many different potential topics and problems. [4] reports extraction accuracies of up to 88% (take these results with a grain of salt).

References

[1] Regina Barzilay. *Text Summarization*, lecture slides, MIT, December 2005. http://people.csail.mit.edu/regina/6864/slides/lec22-4.pdf

[2] Professor Sarkar did not specify whether it was her Master's or PhD thesis, and both seem useful:
Regina Barzilay. *Lexical chains for summarization*. Master's thesis, Ben-Gurion University, 1997.
Regina Barzilay. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation*. PhD Thesis, Columbia University, 2003.

[3] Güneş Erkan and Dragomir R. Radev. *Lexrank: Graph-based centrality as salience in text summarization*. Journal of Artificial Intelligence Research (JAIR), 2004.

[4] E. Barzilay and L. Lee. *Catching the drift: Probabilistic content models*

with applications to generation and summarization. In proceedings of HLT-NAACL 2004 (2004), pp. 113-120.

[5] L. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, 77(2):257-286, 1989.

[6] Kupiec, J., Pedersen, J. and Chen, F. (1995). *A trainable document summarizer*. Proceedings of the 18th ACM SIGIR Conference (pp. 68–73). Seattle.