Skim reading of audio information

Arne Jönsson
Santa Anna IT Research Institute AB
Linköping, Sweden
arnjo@ida.liu.se

Mimi Axelsson, Erica Bergenholm
Bertil Carlsson, Gro Dahlbom
Pär Gustavsson, Jonas Rybing
Christian Smith
Linköping University
Linköping, Sweden

1 Introduction

The objective of this research is to improve browsing of audio information services by providing skim reading at various levels and to present summaries of texts. Summarizing techniques can already be quite useful when applied to written text, but with such a static, visual media you still have the option of skimming through the area in any direction and at any pace you would like. Audio, however, is a strictly linear, non-static media where you can only go in one direction while gathering information. Since most of us are able to make out words, if they are read to us in normal or a little above normal pace, the shortened material produced by a summariser would be immensely useful to people skimming an audio file

Skim reading involves various techniques depending on situation, user and information content. For instance, a politician driving and skim reading a report before a meeting would probably prefer a short summary of the report, whereas a visually impaired person listening to the daily newspaper might prefer keywords reflecting the content of a story.

2 CogSum

We have implemented a summarizing prototype program, named CogSum. We use Random Indexing (Sahlgren, 2005), to create the summaries that best resembles the original content, cf. (Hassel, 2007). Summaries can be isolated words or whole sentences, see below. Using whole sentences means that no speech synthesis is needed, instead sentences are assembled from the original corpus and the sound files are delivered. Vector space techniques used in a similar way on words provide content information. In such cases pre-processing is used to remove stop words, build term lists (with synonyms), and perform stemming. Finally, speech synthesis is needed to deliver the information. CogSum is programmed using Martin Hassel's Java libraries (JavaSDM) for random indexing and as a basis for preprocessing the text that is to be summarized. Stemming is done using the Snowball stemmer (`http://snowball.tartarus.org/`). CogSum also uses PageRank to rank the sentences, as an assurance that the relevant text content would be selected (Chatterjee and Mohan, 2007).

Index vectors of words juxtaposed to the active focus word are collected to create the context vector of the focus word. In addition, words' context vectors are weighted based on the distance from the active focus word, using a weighting scheme represented as a vector. A total document vector is created by adding all unique words' context vectors. The document vector is divided by the total number of unique words, thus creating an average contextual theme vector of the document. The sentences in the document were identified through a search for certain patterns, such as "punctuation-space-capital letter", or "punctuation -space-new paragraph". The program calculates the sentence vector by subtracting the document's average vector from each word's context vector in the current sentence. This follows Chatterjee and Mohans (Chatterjee and Mohan, 2007) theoretical arguments where the average term vector can be seen as the document's central theme.

When the text was indexed, a method to rank the importance of each sentence or word in the document was needed; a similarity measure. The similarity measure of the context vectors used was based on the method to determine vector similarity by using the cosine angle. In order to rank sentences by means of cosine comparison we used PageRank, cf. (Chatterjee and Mohan, 2007). The PageRank algorithm works on a graph, where sentences represent nodes and where the edges are represented by a link between sentences. The weights applied to the link between the nodes are calculated using the cosine comparison between them. By using PageRank the sentences with similar content will therefore in a way “vote” on each other. The sentences which are important for the document will have their values increased more than those that are less important, effectively separating important sentences from the rest. The sentences are then sorted in a descending order upon presentation with their position in the text unaltered.

A variety of features were implemented in the prototype interface; the possibility to choose whether the summary should be presented in key words and/or as sentences, as well as a slide bar to interactively select the level of summarization in percentage. The summary will appear in a separate “pop-up window” where a change in the slide bar immediately affects the text displayed. The user can thus always get feedback on how the summary grows and shrinks as the slider is altered The size of the documents that CogSum summarized was about 200-450 words. The context vectors were weighted so that the words that were right next to focus word got more impact than those who were a little further away. More specifically, the words that were right next to the focus word were weighted to 1, and those that were two steps away was weighted to 0.5.

3 Evaluations

We have conducted two experiments. The first investigated whether people preferred summaries built on keywords or whole sentences for the purpose of skim reading. In the study, 20 students, between 20-30 years old, not visually impaired, listened to sound files of either complete sentences or words of manually created summaries. Humans, not synthetic speech, were used to produce the sound files. The subjects were presented a varying number of sentences, or keywords, representing the ”best” 10, 25, 50 or 70% of the total number of sentences. Stop words were removed when keywords were presented to the subjects. The order in which sentences, or words, occurred in the original text were preserved when presented to the subjects. Our results show that subjects prefer whole sentences to words on all four levels (10, 25, 50 or 70%). Note that in the instructions we informed our subjects that 10% should be seen as a way of deciding if an article is worth reading whereas 70% should be seen as a summary of the text. One could assume that on the 10% or 20% level subjects would prefer words to sentences as an indication to whether the article is interesting to listen to or not, but that was not the case.

Secondly we wanted to see whether a summary created by extracting important sentences from a source presented as audio would pass as skim reading on auditive media, such as audiobooks. To test this we conducted experiments where 30 subjects, different from the 20 used in the word vs sentence experiment, were listening to summaries on the sentence level, answer fact questions on the content of the article and answer questions on the usefulness of the summaries. The summaries were created using CogSum, and then read by the speech synthesizer ”Elin” from InfoVOX. The summaries were split in three categories; headline, 10% summary, and 50% summary. The 10% summary were supposed to act as an indicative summary, giving the subject enough information to determine just the theme or vague aspects of the content, whereas the 50% summary would be sufficient to deliver the most important content and facts of the source, in this case newspaper articles. The experiments show that summaries on the 50% level provide a very good approximation of the whole article. We also found (20 out of 30 subjects) that the 50% summaries gave sufficient information not to have to read the whole article. Furthermore, the 10% summaries gave more information than the headline alone.

References

Nilhadri Chatterjee and Shiwali Mohan. 2007. Extraction-based single-document summarization using random indexing. In *Proceedings of the 19th IEEE international Conference on Tools with Artificial intelligence – (ICTAI 2007)*, pages 448–455.

Martin Hassel. 2007. *Resource Lean and Portable Automatic Text Summarization*. Ph.D. thesis, ISRN-KTH/CSC/A–07/09-SE, KTH, Sweden.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005.*