Generating slides from HTML pages

Arun JVS
201001079

Rohit Girdhar
201001047

Sudheer Kumar
201001149

***Abstract*—The abstract goes here.**

I. Introduction

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.7 and later. I wish you the best of success.

II. Related Work

There has been limited work that addresses the problem of automatic slides generation. Previous works include approaches like section-wise summarization [2], alignment methods for matching document regions with presentation regions [1], extracting topics and itemized elaborations from tagged documents. Many of these systems use multiple ideas from Information Retrieval, such as TF-IDF weights, query expansion, query-specific summarization, POS-tagging, etc.. Some of them are purely generative, while others learn models from an existing corpus of document-presentation pairs. These systems are tuned to work with different input formats such as PDF, XML, LaTeX and PPT, each of which preserve a varying amount of semantic and structural information about the original text. In the following two subsections, we review two papers [2],[1] which particularly deal with technical papers.

A. *SlidesGen*

The first work we review is by Sravanthi et al. [2]. They propose an novel framework for automatic generation of presentation slides for technical papers. They take LaTeX documents of research papers as input, and return the presentation slides. Their method depends on the assumption that by and large, conference papers have a similar structure: an abstract, followed by sections that can be broadly classified as introduction, related work, actual work (model), experiments, conclusion/results and bibliography. A slide in their system contains a title and some bulleted points that are important in that section. They evaluate the system by surveying the response of people who use their system.

Their system is divided into multiple stages. The first stage is pre-processing stage. In this step, the LaTeX documents are converted into XML using a public domain converter LaTeXML.

Next, the generate what they call ”configuration file”, which contains configuration parameters for each section (since each section has a different point of view and writing style). This involves categorization of the section and extraction of key phrases from the section which are then stored in the configuration file. For example, a section with large number of cite tags, or with title containing words such as ”related work” or ”literature survey” is categorized as related works section.

The next step is of extracting key phrases. Most research papers have associated key phrases that can be help categorize the content in paper, and contain important concepts introduced in the paper. They are mostly related to model and experiments section, and can be used to summarize the same. So, the keywords given at the beginning are added to keyphrases for those sections in the configuration file. Also, few other phrases as the names of subsections etc are also added to the configuration file.

Next, they use QueSTS summarizer to summarize the model, experiment and conclusions section. QueSTS represents the text as a Integrated Graph where sentence is a node, and edge exists between 2 sentences if the sentences are similar. The edge weight is defined as the cosine similarity between the 2 sentences (above a given minimum threshold) Given the keyphrases computed previously, they are tokenized and a centrality based (to the query phrase) node weight is computed for the node corresponding to each token. Thus, they get query specific node weights and query independent edge weights.

From the above graph, they construct a Contextual Tree for each term q and node r. All the trees at node r are merged into a Summary Graph at r. The SGraphs generated from each node are ranked using a scoring model, and the one with best rank is returned as the summary.

The final step of this procedure is slide generation from the XML and configuration file. The introduction slides are generated by comparing introduction sentence with the abstract using consine similarity and ones with high similarity are placed in the slides. They give a similar method for generating slides from related work section as well. From model and experiments section, the slides are generated using the above QueSTS mechanism. Similarly, conclusion slides are generated using comparison with keywords such as ”proposed”, ”concluded” etc.

Another important aspect in slides is graphics. Graphics are added along with sentences that either refer to it or is present along with it.

The paper finally discusses the issues in alignment of sentences and generation of slides. They also evaluate their slides manually with multiple users, taking their feedback. Most users gave a satisfaction level of more than 8/10 to the slides generated.

B. *Alignment Methods*

This paper by Brandon et. al. [1] takes a different approach to slide generation, inspired by the human throught process while making a slide out of a paper. They focus their efforts on *alignment* methods - first breaking up the document and presentation into regions and then performing a matching on them. *Slide regions* include bullets, headings, and other text spans, while *Paper regions* include paragraphs, section headings, and list items.

They generate their corpus of 296 paper-presentation pairs from workshops of technical conferences, through simple searching. The papers were PDF format, and presentation were a mixture of PDF and PPT. Before working with them, they convert them into custom XML formats, which represent relevant parts of the original data as logical regions (orthographic boundaries). They prefer such physical regions over semantic regions for its simplicity to implement and verify.

The alignment problem now reduces to an IR query, where the query is a slide region (which is the section/subsection the slide is related to), and the documents are the target regions from the paper. They compare two TF-IDF based scoring methods, with and without query expansion, resulting in 4 alignment methods.

The procedure of scoring is as follows. For each token in each region TF-IDF is computed, where TF is the frequency of the token in the region and DF is the number of regions containing the token's stem. The slide region is tokenized and POS tagged to remove non-content words. Each token in the query is stemmed and then may or may not be query-expanded depending on the method. A score is then calculated for each target region with the query. Two scoring methods were used - one uses the average TF-IDF score of the search terms relative to the target region, and the other uses the quantity of the matched terms.

The evaluation of their methods gives critical insights. First, a vast majority of the slide regions are not alignable (zero score with all target regions) - meaning that a lot of information in slides is not present in the paper - contrary to their hypothesis. Then they define an *alignable* accuracy against the *raw* accuracy, considering only alignable slide regions. They find that the best algorithm on an average gives an average 75% alignable accuracy, but only 50% raw accuracy. Query expansion seems to have little or negative impact on the aligners and that the second scoring method is better than the first.

After more results, they conclude that the data indicates that the task of presentation generation is highly dependent on the end purpose the presentation will serve, as well as the target audience and other factors. Also query expansion generally degraded performance, possibly because authors tend to use wording in slides similar to their paper, and that using synonyms for query expansion is not aggressive enough, and may require hypernyms, immediate hyponyms, and other semantically related terms. Also a possible loss of accuracy could be polysemy of words, causing query expansion to be incorrect and insensitive to context.

References

[1] Brandon Beamer and Roxana Girju. Investigating automatic alignment methods for slide generation from academic papers. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 111–119, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[2] M. Sravanthi, C. Ravindranath Chowdary, and P. Sreenivasa Kumar. Slidesgen: Automatic generation of presentation slides for a technical paper using summarization. In H. Chad Lane and Hans W. Guesgen, editors, *FLAIRS Conference*. AAAI Press, 2009.