University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization:	Spring semester, 20..... Open / Restricted access
Writer:	…… (Writer's signature)
Faculty supervisor: External supervisor(s):	
Thesis title:	
Credits (ECTS):	
Key words:	Pages: …… + enclosure: …… Stavanger, …….. Date/year

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Automatic Generation of Presentations for Research Papers

Master's Thesis in Computer Science

by

Tor Olav Stava

Internal Supervisors

Krisztian Balog

Petra Galuscakova

External Supervisors

External Supervisor 1

External Supervisor 2

Reviewers

Reviewer 1

Reviewer 2

February 5, 2024

"Programming is a nice break from thinking."

Leslie Lamport

Abstract

Acknowledgements

I would like to thank my supervisors for their fantastic enthusiasm and help with writing this thesis.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

The process of creating a presentation from a research paper can be a time consuming and tedious task. The goal of this project is to automate this process, by creating a system that can automatically generate presentations from research papers.

1.2 Objectives

The main objective of this thesis is to create a system that can automatically generate presentations from research papers. In detail, the research papers will be in LaTeX format and the presentations will be in LaTeX Beamer format. In order to achieve this objective, the following research questions will be addressed:

- RQ1: Can we generate a dataset of research papers and their presentation in the required LaTeX and Beamer formats?
- RQ2: Can we provide a method for extracting the relevant information from a research paper?
- RQ3: Can we provide a method for summarizing the extracted information?
- RQ4: Can we provide a method for generating a coherent and high-quality presentation from the summarized information?

1.3 Approach and Contributions

Creating a presentation from a research paper can be viewed as either a summarization task or an extractive task, or a combination of both. We propose a novel method, **REP2BEAM** of generating presenation slides in LaTeX Beamer format from research papers in LaTeX format. The system will extract the relevant information from the research paper and then summarize it in a presentation. The system will be evaluated on how well it can extract the relevant information from the paper and how well it can summarize it in a presentation.

1.4 Outline

The rest of this thesis is structured as follows. Chapter 2 discusses relevant work related to the tasks of summarizing, extracting, and generating presentations based on documents and reports. Chapter 3 describes in detail the approach developed in this thesis in order to create the Automatic Generation of Presentations for Research Papers (**AGPRP**)/**REP2BEAM** system (naming to be decided). Chapter 4 evaluates the results of the AGPRP system by looking av various metrics. Chapter 5 wraps up and concludes the thesis.

Chapter 2

Related Work

2.1 Automatic slide generation of scientific papers

Sefid and Wu [1] presents a novel approach to automatically generate presentation slides from scientific papers, addressing the time-consuming process of slide preparation for researchers. The paper outlines the challenges of summarization, distinguishing between abstractive and extractive methods, and focuses on the latter due to its grammatical correctness and consistency with the original document. The main contributions include the use of deep neural models for sentence encoding within the context of sentence ranking for slide generation and the combination of regression with integer linear programming (ILP) for selecting salient sentences to create bullet points.

The paper reviews related work in text summarization, highlighting the gap in research regarding automatic slide generation. It then details the proposed model, which trains on a dataset of paper-slide pairs to score sentences for importance using convolutional neural networks (CNN), gated recurrent units (GRU), and long short-term memory (LSTM) networks, along with pre-trained word embeddings. This model also incorporates surface features of sentences, such as section placement and sentence length, to aid in scoring.

Further, the process of sentence selection is elaborated, comparing greedy methods with ILP to optimize sentence choice based on salience scores. For slide generation, the model focuses on generating first-level bullet points from key noun phrases in selected sentences, with the aim of creating concise and informative slides.

The evaluation of the model is conducted on a dataset of 1200 scientific papers and their corresponding presentation slides, using the ROUGE metric for assessing the quality of generated summaries in comparison to the original slides. The results show that the model outperforms existing baselines in terms of ROUGE scores, indicating a higher

overlap with manually created slides. The dataset does not seem to have been made publicly available, but the same authors have published a larger dataset in a later paper [2].

In conclusion, the report underscores the effectiveness of combining feature-based and deep learning approaches for slide generation from scientific texts. It suggests that further improvements could be made by integrating visual elements into slides, which were not covered in this study. The implementation of the model is made available on GitHub for further research and development in this area.

2.2 DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents

Fu et al. [3] introduces DOC2PPT, a novel system designed to automate the creation of presentation slides from scientific documents. The system is tasked with understanding and summarizing complex documents that contain both textual and visual elements, thus requiring advanced vision-and-language processing capabilities. Traditional approaches to document summarization and cross-modal retrieval fail to address the unique challenges of slide generation, which include the need for multimodal summarization, structured output generation, and optimal visual layout design. To overcome these challenges, the authors propose a hierarchical recurrent sequence-to-sequence architecture that processes the input document at the section level and generates a structured slide deck. This architecture includes a document reader for encoding sentences and figures, a progress tracker to manage the flow of content from document sections to slides, an object placer for determining the placement of textual and visual elements on slides, and a paraphraser to convert document-style sentences into concise, slide-friendly text.

The report details the creation of a new dataset comprising 5,873 pairs of scientific documents and corresponding slide decks, ref. Table 2.1, which were used to train and evaluate the DOC2PPT system. Several novel quantitative metrics are introduced to measure the quality of the generated slides, including Slide-Level ROUGE (ROUGE-SL) for textual content, Longest Common Figure Subsequence (LC-FS) for figure content, Text-Figure Relevance (TFR) for the relevance of text to figures, and mean Intersection over Union (mIoU) for layout quality.

The experimental results demonstrate the effectiveness of the hierarchical modeling approach, with significant improvements observed across all evaluation metrics compared to baseline models. The incorporation of a paraphrasing module and a text-figure matching objective further enhances the quality of the generated slides, both in terms of

textual conciseness and the relevance of figures to text. Human evaluation also confirms the superiority of the DOC2PPT system over simpler models and baseline approaches.

In conclusion, the DOC2PPT system represents a significant advancement in the automatic generation of presentation slides from scientific documents, offering new opportunities for human-AI collaboration in the preparation of academic and research presentations. The large dataset introduced alongside the system, as well as the novel evaluation metrics, are expected to spur further research and development in the vision-and-language domain.

2.3 ArXiv Table Extractor

The project "arXiv Table Extractor" developed by Ramsay and Pop [4] aimed at automatically extracting and parsing tables from scientific papers uploaded to arXiv. Utilizing a combination of Python, Docker, MongoDB, and various Python libraries, the team developed an automated system capable of downloading papers, extracting tables from LaTeX source files, and parsing these tables into a structured JSON format. The system was designed to run as a service, potentially hosted on a server, to automate the process of table extraction and parsing, making the data easily accessible and usable for further analysis or visualization.

During the Minimum Viable Product (MVP) phase, the project showed a strong ability to extract tables that did not contain multi-column or multi-row formats, achieving a high accuracy rate in those instances. However, it struggled with tables that utilized these more complex LaTeX constructs, resulting in a significant portion of tables being unextracted or incorrectly parsed.

Further Development 1 (FD1) addressed some of these issues, particularly improving the handling of multi-column tables. This led to a substantial increase in the success rate of table extraction, though challenges remained with very complex tables and those missing certain LaTeX formatting elements like the end-of-table indicator (\\).

Future improvements suggested include enhancing the extraction and parsing algorithm to handle a broader range of table formats, including those with multi-row formats and tables missing LaTeX indicators. Additionally, the development of a user-friendly front-end interface was proposed to make the extracted data more accessible to users, suggesting a need for a backend to manage data retrieval from the database.

The project demonstrates a successful application of automated document parsing to scientific literature, with significant potential for expansion and refinement. The ability

to accurately extract and parse table data from a vast repository like arXiv could significantly aid in data analysis and research across various scientific fields.

2.4 Extractive Research Slide Generation Using Windowed Labeling Ranking

Sefid et al. [2] introduces a method for automatically generating presentation slides from academic papers by selecting salient sentences to form bullet points. This approach aims to reduce the time and effort required to manually create slides. The primary challenge lies in accurately extracting key points from an academic paper, considering the limitations of existing methods in fully understanding sentence semantics and their interrelations. The proposed solution involves an extractive summarizer that identifies important sentences within consecutive sentence windows using neural networks. These selected sentences, along with their frequent noun phrases, are then organized in a layered format to create slide bullet points.

Presentation slides typically consist of multi-level bullet points, with the majority of content found in the first and second levels. Thus, the developed system focuses on generating slides with up to two levels of bullet points. The contribution of this work is threefold: proposing a system for slide generation from high-ranking sentences, creating a corpus of 5,000 paper-slide pairs (PS5K) in computer and information science, and introducing a novel method for ranking sentences within a window, significantly improving upon existing text summarization methods.

The paper also reviews related work in the area of automatic slide generation from scientific papers, highlighting differences from standard text summarization and the limitations of previous approaches. The newly proposed model was trained and evaluated on the PS5K dataset, demonstrating superior performance in identifying key content for slides. This model processes documents through steps including sentence labeling based on semantic similarity to slides, embedding sentences and documents for ranking, and selecting sentences for inclusion in slides.

Experiments conducted on the PS5K dataset explored various configurations, including window sizes for sentence selection, showing that a window size of 10 yielded the best results in terms of ROUGE-1 recall. The study utilized Stanford CoreNLP for tokenization, lemmatization, and noun phrase extraction, and GloVe vectors for word embeddings. The summaries generated by the proposed method were evaluated using standard ROUGE scores, outperforming base models and highlighting the effectiveness of distributing positive labels across a paper's sections for summarization.

Source	Number of Report/Presentation Pairs	Reports Format	Presentations Format	Size
Fu et al. [3]	5873	PDF	JPEG	33.8 GB
Sefid and Wu [1]	5000	PDF	PDF	16 GB

Table 2.1: The number of papers and presentations in the dataset.

In conclusion, this work contributes significantly to the field of automatic slide generation for scientific presentations by providing a large dataset for training and evaluation, proposing an efficient method for summarizing scientific articles, and demonstrating the importance of considering the hierarchical structure of academic papers in the summarization process.

2.5 Available Datasets

Only two datasets have been identified that are available online and that are suitable for training and evaluating the automatic generation of presentations from research papers. The datasets by Fu et al. [3] and by Sefid and Wu [1] are summarized in Table 2.1. The datasets are not directly applicable to the proposed task in this thesis, since the papers should be in LaTeX format and the presentations should be in LaTeX Beamer format. However, it may be possible to use the datasets as a basis for curating or creating a new dataset that is more suitable for the task at hand. It may also be possible to use existing conversion tools to convert the papers and presentations to the desired format.

Chapter 3

Approach

3.1 Overview

The Automatic Generation of Presentations for Research Papers (AGPRP) is a system for generating presentations in LaTeX Beamer format from academic research papers. It is composed of three main components: a report parser, a presentation generator and a presentation compiler. The report parser is responsible for extracting the relevant information from the report, the presentation generator is responsible for generating the presentation and the presentation compiler is responsible for compiling the presentation into a LaTeX Beamer file. The system is designed to be modular, so that each component can be replaced with a different implementation. This allows for easy extension of the system, as well as for easy testing of different approaches.

A two-step approach is suggested for generating the slides: first, the high-level topics are identified and used to generate slide titles, then bullet points are generated for each slide. An optional third step is to generate visual content for the slides, such as tables and illustrations.

3.1.1 NOTES (Section to be deleted)

- Could it be beneficial to use a knowledge graph to represent the information in the report? This could be used to generate the presentation, as well as to provide a visual representation of the information in the report.
- Summarizing vs extracting information from the report. How do we decide the best approach?
- Should we test various models, e.g. HuggingFace transformers vs ChatGPT?

- Testing how well already existing models (e.g. ChatGPT, MS Copilot) perform when asked to generate a presentation from a research paper.
- When using LaTeX as source file, how do we identify the main file? Where do we start parsing?
- Extract outline from PDF/PPT titles, add additional metadata (e.g. bullets, tables, illustrations), generate slides from outline. (Few-shot prompting LLM?) (Manually procure a high-quality dataset)

3.2 Dataset

A major contribution of this thesis is the creation of a dataset of research papers and their corresponding presentations. The dataset will be used to train and evaluate the system. The dataset will be created by collecting research papers and their corresponding presentations from various sources. The dataset will be in LaTeX format for the reports and in PDF/PPT format for the presentations. Ideally both the reports and the presentations should be in LaTeX format, but correspodning presentations in the wanted format has proven challenging to find in practice.

Some datasets are already available as shown in chapter 2, but they are not in the format we need. However, they can be used as a starting point for creating the dataset by using the presentations and mining the corresponding research papers in the required format. The arXiv archive is a good source for research papers, and the dataset can be created by mining the arXiv archive for research papers that are avaialable in LaTeX format. It's important to adhere to the copyrights for the scraped data if we want to publisize the dataset. We also need to follow the data scraping guidelines put forth by arXiv.

3.2.1 Data Collection

3.2.2 Data Preprocessing

If we need to work with PDF formats, there will be a need to convert them to LaTeX format before we can train our model on the dataset.

3.2.3 Data Augmentation

It may be useful to enrich the dataset with details such as internal and external references, and tables and illustations. This can help when generating the presentation, as it can be used to determine which additional items to include together with the text.

3.3 Tools

3.3.1 LaTeX

Parsing LaTeX documents may be done using the TexSoup Python library[1].

3.3.2 PDF

There are several tools that may be able to convert PDF files into text or LaTeX format:

- https://github.com/kermitt2/grobid - "A machine learning software for extracting information from scholarly documents"
- **PDFMiner**[2] - PDFMiner is a tool for extracting information from PDF documents. Unlike other PDF-related tools, it focuses entirely on getting and analyzing text data. PDFMiner allows one to obtain the exact location of text in a page, as well as other information such as fonts or lines. It includes a PDF converter that can transform PDF files into other text formats (such as HTML). It has an extensible PDF parser that can be used for other purposes than text analysis.
- **pdftolatex**[3] - "Python tool for generation of LaTeX code from PDF files."
- **pdf2latex-converter**[4] - "Originally based on **pdftolatex**." (Work in progress).
- **pdf2latex**[5] - "pdf2latex is a CLI tool to convert a PDF back to LaTeX."
- **pdf2latex**[6] - "Train a neural network to produce latex source code which generates a given pdf file."

[1] https://texsoup.alvinwan.com/
[2] https://pypi.org/project/pdfminer/
[3] https://github.com/vinaykanigicherla/pdftolatex
[4] https://github.com/mcpeixoto/pdf2latex-converter
[5] https://github.com/emsquid/pdf2latex
[6] https://github.com/safnuk/pdf2latex

- **PDF2LaTeX**[7] - PDF2LaTeX is a tool for converting PDF files into LaTeX format. It is a command-line tool that can be used to convert PDF files into LaTeX format. It is written in Python and uses the PyPDF2 library to parse PDF files. It can be used to convert PDF files into LaTeX format.
- **pypdf**[8] - pypdf is a Python library for working with PDF files. It can be used to extract text from PDF files, as well as to convert PDF files into other formats (such as HTML).
- **pdfly**[9] - "CLI tool to extract (meta)data from PDF and manipulate PDF files."

3.4 Method

3.4.1 Report Parser (RP)

The *Report Parser (RP)* is responsible for extracting the relevant information from the report. The report parser is composed of two main components: the *Report Content Extractor (RCE)* and the *Report Content Summarizer (RCS)*. The *Report Content Extractor* is responsible for extracting the relevant information from the report, while the *Report Content Summarizer* is responsible for summarizing the extracted information.

3.4.2 Presentation Content Generator

The *Presentation Content Generator (PCG)* is responsible for Presentations can be generated in a wide range of variations in layout, content and style.

3.4.3 Presentation Slides Generator

There are several options when it comes to generating visual layout and content for the presentation slides.

To keep things simple we propose three simple slide layouts: A title slide, a bullet point slide and a visual slide. The title slide will contain the title of the slide, the bullet point slide will contain a list of bullet points and the visual slide will contain a visual representation of the information in the report.

[7] https://github.com/senyalin/PDF2LaTeX
[8] https://pypi.org/project/pypdf/
[9] https://github.com/py-pdf/pdfly

Chapter 4

Experimental Evaluation

Page budget for Evaluation: 10-15 pages

- Detail your evaluation methodology, present your results, and provide an analysis of them. Results can be quantitative and/or qualitative (from benchmark, user study, user satisfaction survey, etc.).
- It is strongly desired that you have empirical results, nevertheless, this may not be applicable to all types of theses.

4.1 Experimental Setup

- Explain the methodology used for evaluating your contribution, and the metrics used for evaluation.
- If you use any dataset, explain it, detail its version, and mention briefly some main statistics about it, of interest for your problem (e.g., size, provenience, etc.), if appropriate.
- If you collect ground truth data, describe your annotation experiment. Explain what the annotators were asked to do (and show a screenshot or schema if available). Detail the number of annotators, their nature (experts, or crowdworkers), the criteria for deciding on each annotation instance (e.g., majority class, dynamic judgments, etc.), the criteria for ensuring quality (e.g., minimum accuracy, filters). If possible, report the inter-annotator agreement coefficient and mention how strong this value means that the agreement is.

4.2 Experimental Results

- Present the results, using tables and (pretty) plots.

4.3 Analysis

- Now that you presented the results, what do these results actually mean (esp. regarding the objectives you set out in the introduction)?
- Can you identify success and failure cases?
- What do the results say for individual parts you evaluate and overall in combination?
- Make sure you formulate clear take-home messages.

Chapter 5

Conclusions

Page budget for Conclusions: 3-5 pages

- Summary of the work you have done, what worked and what didn't
- Make sure it connects well with the Introduction, by answering every RQ.

5.1 Future Directions

Discuss potential future work that may fill gaps in your work, or approaches that seem promising to overcome problems you encountered but that you weren't able to tackle.

Bibliography

[1] Athar Sefid and Jian Wu. Automatic slide generation for scientific papers. In *Third International Workshop on Capturing Scientific Knowledge co-located with the 10th International Conference on Knowledge Capture (K-CAP 2019), SciKnow@ K-CAP 2019*, 2019.

[2] Athar Sefid, Jian Wu, Prasenjit Mitra, and Lee Giles. Extractive research slide generation using windowed labeling ranking, 2021.

[3] Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yang Song. DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents. *Proceedings of the ... AAAI Conference on Artificial Intelligence*, 36(1):634–642, 2022.

[4] David Gordon Ramsay and Rebeca Pop. Arxiv table extractor. Bachelor's thesis, University of Stavanger, 2021.