Extractive Research Slide Generation Using Windowed Labeling Ranking

Athar Sefid
Penn State University
atharsefid@gmail.com

Jian Wu
Old Dominion University
j1wu@odu.edu

Prasenjit Mitra
Penn State University
pum10@psu.edu

Lee Giles
Penn State University
clg20@psu.edu

Abstract

Presentation slides generated from original research papers provide an efficient form to present research innovations. Manually generating presentation slides is labor intensive. We propose a method to automatically generates slides for scientific articles based on a corpus of 5000 paper-slide pairs compiled from conference proceedings websites. The sentence labeling module of our method is based on SummaRuNNer, a neural sequence model for extractive summarization. Instead of ranking sentences based on semantic similarities in the whole document, our algorithm measures importance and novelty of sentences by combining semantic and lexical features within a sentence window. Our method outperforms several baseline methods including SummaRuNNer by a significant margin in terms of ROUGE score.

1 Introduction

Nowadays it is a common practice for researchers to use slides as a visual aid to present research findings and innovations. Slides usually contain bullet points that the researchers believe to be important to show. Manually creating a set of high-quality slides from an academic paper is time-consuming. We propose a method that automatically selects salient sentences that could be included into the slides, with the purpose of reducing the time and effort for slide generation. The main challenge towards solving this problem is accurately extracting the main points from an academic paper. In this paper, we propose an extractive summarizer that identifies the best sentence in a set of consecutive sentence windows. The selection process depends on importance and novelty of the sentence modeled by neural networks. The selected sentences and their frequent noun phrases are structured in a layered format to make the bullet points of the slides.

Our contribution is threefold.

- We proposed a system that utilizes sentences with high ranks for generating presentation slides for research papers, which can be used as a starting point in the slide generation process.
- We provide PS5K, a corpus of 5000 paper-slide pairs in the field of **computer** and **information science**. To our best knowledge, this is the largest paper-slide dataset that could be used for training and evaluating slide generation models.
- We proposed a novel method to rank sentences within a sentence window, which improved an existing state-of-the-art text-summarization method by a significant margin.

2 Related Work

Summarizing scholarly articles in presentation slides is different from standard text summarization (Xiao and Carenini, 2019), which focuses on generating a paragraph of free text summary out of a long document. Automatic slide generation can be achieved by first extracting salient sentences in a hierarchical order and grouping them into slides that are sequentially aligned with the original paper.

PPSGen was a framework that automatically generates presentation slides from scientific papers (Hu and Wan, 2014). They applied Support Vector Regressor and Integer Linear Programming (ILP) to rank and select important sentences. Wang et al. (2017) generate slides by extracting phrases from papers and learning the hierarchical relationship between pairs of phrases to build the structure of bullet points. Their model is trained on a small set of 175 paper-slide pairs. The slideSeer (Kan, 2007) project crawled more than 10,000 paper-slide pairs using the Google APIs to search for the slide of papers using their title as a search query. The full set of data is not publicly available (only 20 pairs are available). Compared with previous works, our

model is trained and tested on a relatively large set of 5000 paper-slide pairs and the dataset will be publicly available for future works.

SummaRuNNer (Nallapati et al., 2017) is a neural extractive summarizer that treats the summarization task as a sequence labeling problem. SummaRuNNer was evaluated on CNN/Daily Mail corpus, which contains news articles that are shorter than research papers. We improve the SummaRuNNer model to suit for summarization of scientific papers.

3 Data

Producing a large dataset for summarization of scientific documents is challenging and it requires domain experts to make the summary. The latest CL-Scisumm 2018 summarization task contains **only 40 NLP papers** with human-annotated reference summaries. Recently, ScisummNet (Yasunaga et al., 2019) expanded the CL-Scisumm to 1000 scientific articles. Using presentation slides made by the authors is promising for the training of deep neural summarization models as more conferences are providing slides with papers.

We crawled more than 5,000 paper-slide pairs from a manually curated list of websites, e.g., usenix.org and aclweb.org. GROBID (Lopez, 2009) is used to get metadata and the body of the text from scientific papers in PDF format. Presentations are transformed form PDF or PPT format to XML by Apache Tika[1]. The Tika XML files are divided into *pages* and the text is extracted using Optical Character Recognition (OCR) tools. Most venues of papers in our dataset are in computational linguistics, system, and system security. In our dataset, there are on average 35 pages of slide per presentation and 8 lines of text per slide page. The majority (75%) of papers are published between 2013 and 2019. We used this dataset (called PS5K) to train summarization models to identify important parts of the input document at the sentence level. The dataset is available here.

4 Method

Generating slides requires identifying important sentences. It starts with the labeling and ranking of salient sentences and ends with extraction of frequent noun phrases as bullet points. The architecture of our model is shown in Figure 1

[1]https://tika.apache.org/

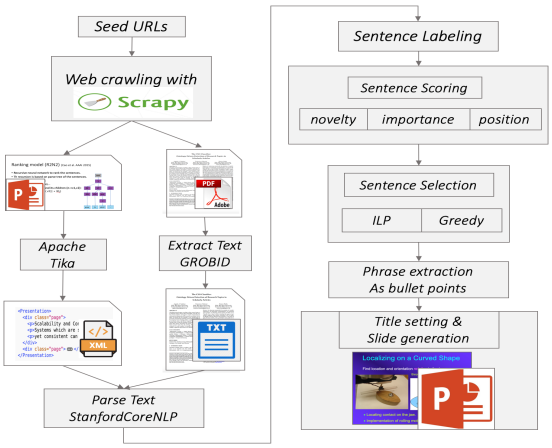

Figure 1: The main components of the model for summarizing the paper and building the slides.

4.1 Sentence Labeling

The text in human generated slides may not be directly extracted from the original paper. Instead, text can be truncated, summarized, or rephrased. Therefore, we treat each slide as an *abstractive* summary of a paper. The sentence labeling process attempts to identify salient sentences that are semantically similar to the corresponding slides. This generates an *extractive* summary, which will be used as the ground truth for training and evaluation. The problem is formalized below.

A research paper can be represented as a sequence of n sentences $D = \{s_1, s_2, ...s_n\}$, each having a label $y_i \in \{0, 1\}$, the system predicts $p(y_i = 1)$, probability of including sentence i to the summary.

SummaRuNNer treats the summarization task as a sequence labeling problem, if adding the sentence to the summary improves the ROUGE score, the sentence is labeled with 1, otherwise it is labeled with 0. This method is suitable for news articles such as CNN/DailyMail (Nallapati et al., 2016) where the first couple of sentences in articles usually cover the main content. Scholarly papers usually contain a hierarchical structure of sections. Each section should have its own summary as a part of the summary of the entire paper. Therefore, the labeling process should be adapted to distribute positive labels across all sections of the paper. However, accurately parsing sections of open domain scholarly papers is non-trivial. Therefore, we propose a windowed labeling approach, in which ranking is performed only within a series of non-overlapping text windows, each of which contains w consecutive sentences. A sentence is

labeled as 1 if adding the current sentence increases the ROUGE-1 index. The best window size is determined empirically by trying different widow sizes and calculating the ROUGE score between selected sentences and the presentation slides. Section 5 elaborates on the experiments performed to select the best window size.

4.2 Sentence and Document Embedding

The ranking of sentences depends on their salience, novelty, and content similarity to the ground truth. To quantify these characteristics, a document is represented into a vector. We explore two methods to build the embedding for the whole document.

Simple Document Embedding A simple document embedding can be obtained by calculating the average of sentence encodings generated by a Bi-directional Long Short-Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997). A sentence s_i can be encoded as $E_{s_i} = [\overrightarrow{h_i}, \overleftarrow{h_i}]$ in which E_{s_i} is a concatenation of forward ($\overrightarrow{h_i}$) and backward ($\overleftarrow{h_i}$) hidden states of the last token in sentence s_i. The embedding for document D with n sentences is the average of all sentence embeddings: $E_D = ReLU(W \times \frac{1}{n}\sum_{i=1}^{n} E_{s_i} + b)$. in which $ReLU$ is the activation function, W and b are parameters to be learned.

Hierarchical Self Attention Document Embedding This model embeds a document by applying the attention mechanism at both word and sentence levels (Al-Sabahi et al., 2018; Yang et al., 2016).

Sentence embeddings are obtained by encoding word-level tokens of a sentence using BiLSTM and then aggregating hidden layers using an attention mechanism. Formally, considering a sentence s_i with m words, the sentence encoding h_{s_i} is obtained as a concatenation of all m hidden states of word-level tokens ($h_{s_i} = [h_1, h_2, ..., h_m]$) where $h_{s_i} \in \mathbb{R}^{m \times 2d}$ and d is the embedding dimension for each word. The attention weights are:

$$a_{\text{word}} = softmax\left(W_{\text{attn}} \times h_{s_i}^T\right) \tag{1}$$

where $W_{\text{attn}} \in \mathbb{R}^{k \times 2d}$ is the model matrix to be learned. Then $a_{\text{word}} \in \mathbb{R}^{k \times m}$ and the embedding for sentence s_i is: $E_{s_i} = average_k\left(a_{word} * h_{s_i}\right)$ where $E_{s_i} \in \mathbb{R}^{1 \times 2d}$ and k is the attention dimension which is set to 100 in our experiments.

Document embeddings (E_D) are generated using sentence embeddings (E_{s_i}) built in the previous step. A similar attention layer is applied on top of sentence embeddings to build the document embedding. The sentence level attention works as the weights to emphasize important sentences in document embedding.

4.3 Sentence Ranking

The rank of a sentence depends on its position in the paper, salience, and novelty with respect to the previously selected sentences, calculated below:

$$\begin{gathered} pos = position \times W_{pos} \\ content = E_{s_i} \times W_{content} \\ salience = E_D \times W_{salience} \times E_{s_i}^T \\ novelty = summary_i \times W_{novelty} \times E_{s_i}^T \\ p(y_i = 1) = \sigma(pos + content + novelty + salience) \end{gathered} \tag{2}$$

where $W_{pos} \in \mathbb{R}^{2d*1}$,where $W_{content} \in \mathbb{R}^{2d*1}$ $W_{salience} \in \mathbb{R}^{2d*2d}$, and $W_{novelty} \in \mathbb{R}^{2d*2d}$ are parameters to be learned. The $position$ is the position of the sentence in the document specified by a Embedding lookup function, σ is the sigmoid activation function, and pos is its positional embedding. The $salience$ estimates the importance of a sentence. The $novelty$ represents the novelty of a sentence with respect to the current summery. The summary embedding is the weighted sum of the previous sentences added to summary until sentence i: $summary_i = \sum_{j=0}^{i-1} p(y_i = 1) \times E_{s_i}$.

With windowed labeling, the positive labels are sparse. To deal with the imbalanced positive labels, the following weighted cross-entropy loss is adopted. The setting of $w_1 = -85$ and $w_2 = -2$ results in the highest ROUGE score.

$$\begin{gathered} -\sum_{i=0}^{n} w_1 y_i \times \log\left(p(y_i = 1)\right) \\ +w_2(1 - y_i) \times \log\left(1 - p(y_i = 1)\right) \end{gathered} \tag{3}$$

4.4 Sentence Selection

To select the sentences for the slide we tried 1) the greedy approach that sequentially adds sentences with highest scores until the maximum limit is hit and 2) the ILP method that selects the sentences by optimizing the following function using IBM CPLEX Optimizer.

$$\begin{gathered} \max \sum_{i \in N_s} l_i x_i \times p(y_i = 1) \\ \sum_i l_i x_i < maxLen, \quad \forall i,\ x_i \in \{0, 1\} \end{gathered} \tag{4}$$

where $p(y_i = 1)$ is the score of the sentence predicted by the model, x_i is a binary variable showing whether sentence i is selected for the summary or

Table 1: Bullet points statistics.

Bullet-Point	Fraction	Avg Word Count
Title	-	3.7
Level 1	56.5%	7.38
Level 2	35.5%	7.22
Level 3	7.9%	6.7

not, l_i is the length of sentence i and penalizes short sentences, and $maxLen$ is the maximum length of the summary.

4.5 Slide Generation

A typical presentation slide includes a limited number of bullet points as the first-level, which are usually phrases or shortened sentences. Some slides may contain second-level bullet points for further breakdowns. Table 1 shows that less than 8% of the content of the presentations in the ground truth corpus is covered in third-level bullets. We generate slides containing up to 2 bullet levels. Table 1 also shows that a slide title on average contains 4 words and either Level 1 or Level 2 bullets contains on average 8 words. Each slide consists of on average 36 words in 5 bullets and each level-1 bullet includes 2 second-level bullets.

Sentences selected are treated as the second-level bullets. The first-level bullets are the noun phrases extracted from the sentences. Noun phrases are removed if they contain more than 10 words or just 1 word. Noun phrases with a document frequency greater than 10 are excluded (e.g. "the model"). The section, which the first sentence of a slide is in, is found and its heading is used as the slide title. The heading is truncated to the first 5 tokens. We limit a maximum of 4 sentences per slide. If a topic has more than 4 related sentences, the slide is split into two distinct ones. A presentation slide generated by our model is available here.

5 Experiments and Results

We estimated the parameters of our model on PS5K. We split the dataset into training, validation, and testing set, each consisting of 4500, 250, and 250 pairs, respectively. We experimented with different window sizes and found that a window size of $w = 10$ gives the best ROUGE-1 recall (Table 2) and is adapted for our model.

The Stanford CoreNLP (Manning et al., 2014) is used to tokenize and lemmatize sentences to the constituent tokens and to extract noun phrases. GloVe (Pennington et al., 2014) 50-dimensional vectors are used to initialize the word embeddings. With the AdaDelta optimizer and a learning rate of 0.1, we trained for 50 epochs. The sentences are truncated or padded to have 50 tokens (only 8% sentences consist of more than 50 tokens). Similarly, we adopt a fixed document size of 500 sentences (only 3.5% of documents in our dataset have more than 500 sentences). We used the standard ROUGE score (Lin, 2004) to evaluate the summaries. The ROUGE scores for summaries are tabulated in Table 3. The summary size can not exceed 20% of the size of the input document in words. TextRank (Mihalcea and Tarau, 2004) is a graph based summarizer that applies the Google PageRank (Page et al., 1999) algorithm to rank the sentences. Sefid et al. (Sefid et al., 2019) rank the sentences by combining surface features, semantic and contextual embeddings. The windowed SummaRuNNer+ILP model outperforms the base SummaRuNNer by at least 3 points in ROUGE-1 recall. Adding attention layer to the model does not improve the ROUGE score while it increases the training time considerably as there are more parameters to be trained.

Table 2: ROUGE scores for oracle summaries generated with different window sizes.

Window Size	ROUGE-1	ROUGE-2	ROUGE-L
3	42.95	11.13	21.59
5	44.34	11.43	22.35
7	44.88	11.64	22.47
10	**45.93**	**12.00**	**22.75**
15	45.52	11.84	22.68

Table 3: ROUGE scores for different models. Oracle and TextRank are unsupervised and do not need training. T_{tr} standards for training time in hours based on Nvidia GTX 2080 Ti GPU. SRNN stands for SummaRuNNer.

Models	R-1	R-2	R-L	T_{tr}
Oracle (window=10)	57.12	16.53	27.62	-
Sefid et al. (Sefid et al., 2019)	36.33	8.73	17.02	-
TextRank (Barrios et al., 2016)	38.87	9.28	19.75	-
SRNN+ILP	45.12	11.65	22.96	18
SRNN+greedy	45.04	11.67	23.03	18
Attn+windowed SRNN+ILP	47.49	11.67	22.89	38
Attn+windowed SRNN+greedy	47.56	11.68	23.30	38
windowed SRNN+ILP	**48.29**	12.00	**23.80**	18
windowed SRNN+greedy	48.28	**12.02**	22.14	18

Conclusion We provide PS5K, which is by far the largest dataset we know, consisting of 5,000 scientific articles and corresponding manually made slides. This dataset could be used for scientific document summarization and slide generation. The code and data will be publicly available.

References

Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. 2018. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212.

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yue Hu and Xiaojun Wan. 2014. Ppsgen: Learning-based presentation slides generation for academic papers. *IEEE transactions on knowledge and data engineering*, 27(4):1085–1097.

Min-Yen Kan. 2007. Slideseer: A digital library of aligned document and presentation pairs. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 81–90. ACM.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Patrice Lopez. 2009. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'09, pages 473–474.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *The SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Athar Sefid, Jian Wu, Prasenjit Mitra, and C Lee Giles. 2019. Automatic slide generation for scientific papers.

Sida Wang, Xiaojun Wan, and Shikang Du. 2017. Phrase-based presentation slides generation for academic papers. In *AAAI*.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3009–3019. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.