Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 2, pp. 181–190, DOI 10.12694/scpe.v24i2.2238

RESEARCH HIGHLIGHT GENERATION WITH ELMO CONTEXTUAL EMBEDDINGS

TOHIDA REHMAN[*], DEBARSHI KUMAR SANYAL[†], AND SAMIRAN CHATTOPADHYAY[‡]

Abstract. With the advent of digital publishing and online databases, the volume of textual data generated by scientific research has increased exponentially. This makes it increasingly difficult for academics to keep up with new breakthroughs and synthesise important information for their own work. Abstracts have long been a standard feature of scientific papers, providing a concise summary of the paper's content and main findings. In recent years, some journals have begun to provide research highlights as an additional summary of the paper. The aim of this article is to create research highlights automatically by using various sections of a research paper as input. We employ a pointer-generator network with a coverage mechanism and pretrained ELMo contextual embeddings to generate the highlights. Our experiments shows that the proposed model outperforms several competitive models in the literature in terms of ROUGE, METEOR, BERTScore, and MoverScore metrics.

Key words: deep learning, pointer-generator network, ELMo, natural language generation.

AMS subject classifications. 68T07, 68T50

1. Introduction. Van Noorden et al.[1] estimated that every nine years the quantity of scientific articles roughly doubles. While the growth in the number of scientific papers is generally viewed as a positive development, it also presents some challenges. With the exponential growth in the number of scientific papers being published [2], it can be challenging for researchers to stay up-to-date with the latest findings in their field and identify which papers are the most important. Although both research highlights and abstracts are summaries of the research article, they serve different objectives and have distinct qualities. The research highlights typically consist of a brief, bulleted list of the main findings of the paper. By providing a concise summary of the most important aspects of the research, the highlights can help readers to quickly assess the relevance of the paper and determine whether it is worth reading in full. The main findings and contributions of the paper can be emphasized in promotional materials such as social media posts by utilizing the highlights.

Text summarization is a technique to prepare a compact text document that has been condensed while retaining its most important and salient information. Text summarization is classified into two types [3]: extractive summarization, which involves picking and combining relevant sentences or phrases directly from the original text [4], but abstractive summarization, which involves generation and formation of new sentences which capture the essence of the original text [5]. In this paper, we propose a method to generate research highlights (a form of summary) of a research paper using deep neural network-based model. Unlike large pretrained language models (often called foundation models [6]) that require access to a huge document corpus, large training time, a huge energy expenditure, our proposed method is task-specific, utilizes pretrained embeddings, and is trained with a much smaller domain-specific corpus. Thus, it is a scalable model suitable for generation of research highlights from scientific papers. The primary contributions of this article are:

1. Our proposition involves integrating ELMo embeddings with pointer-generator networks that utilize coverage mechanisms.
2. We examine how well the proposed model can generate research highlights using two different types of inputs: (a) only the abstract, and (b) a combination of the abstract, introduction, and conclusion of a

[*] Department of Information Technology, Jadavpur University Salt Lake Campus, Kolkata-700106, West Bengal, India. (tohidarehman.it@jadavpuruniversity.in)

[†] School of Mathematical & Computational Sciences, Indian Association for the Cultivation of Science, Jadavpur, Kolkata-700032, West Bengal, India. (debarshi.sanyal@iacs.res.in)

[‡] Department of Information Technology, Jadavpur University, Salt Lake Campus, Kolkata-700106, West Bengal, India. (samiran.chattopadhyay@jadavpuruniversity.in)

research paper.

3. We evaluate our models extensively through multiple metrics, including ROUGE [7], METEOR [8], BERTScore [9], and MoverScore [10] metrics. We show that the proposed model outperforms other existing techniques available in the literature. We also identify the role of each component of our model using an ablation study.

2. Literature survey. Extractive approaches are a text summarization technique that focuses on identifying the most important phrases or sentences from the source text and present them as a summary. Luhn et al.[3] proposed a method of text summarization to select high-scored sentences based on the high frequency words while ignoring the common words. Baxendale et al. [11] proposed using the position of a sentence to select the important sentences of a document. In that research found that 85% of the theme sentences selected from the first sentences of the paragraph and 7% as the last sentence of a paragraphs. Edmundson et al.[12] proposed a method for automatically summarising texts that assigns a score to each sentence based on four features including sentence position, word frequency, document skeleton, and cue words. Kavita et al. [13] proposed a graph based abstractive model called "Opinosis" useful for highly repetitive opinions. The progress of sequence-to-sequence (Seq2Seq) models has significantly improved the state-of-the-art in abstractive summarization [14]. A neural network-based Seq2Seq models used to learn a map of a sequence of input tokens to a sequence of output tokens. Bahdanau et al. [15] improved the fundamental encoder and decoder models' performance. Chopra et al. [16] proposed a unique "Convolutional Attention-based Conditional Recurrent Neural Network (CARCNN)" architecture for abstractive text summarization. On the Gigaword Corpus and the DUC 2004 datasets, the proposed model was evaluated. Nallapati et al. [5] proposed an abstractive text summarization technique that uses "Attentional Encoder-Decoder Recurrent Neural Networks". The authors proposed a model that leverages this architecture to generate a summary of a given input document. Using bidirectional recurrent neural network the model first encodes the input document, which captures the input's contextual information. At the docedoer end, the summary is then generated one word at a time by considering encoded inputted document. The attention mechanism helps the model to concentrate on the key passages of the text when generating the summary. See et al.[17] proposed a model to overcome the problem of out-of-vocabulary words (OOV) and repetition words generation named as pointer-generator network with coverage mechanism. Coverage mechanism helps to avoid repetition by keeping track of what has been summarized by pointing and copy words from the inputted text. Gehrmann et al.[18] proposed a model "Bottom-Up Abstractive Summarization" which enhances the capacity to condense content, while still creating fluent summaries. Liu et al. [19] proposed a transformer-based model for abstractive summary generation of Wikipedia articles. For better word semantic representation, a model combining the pointer-generator model with two pre-trained word embeddings—word2vec and FastText [20].

Scientific paper summarization can be broadly categorized into two types: abstract generation from the paper and summary generation based on the citation [21]. In the past, extractive summarization methods have been widely used for summarizing scientific articles. Kupiec et al.[4] used a limited dataset of 188 scientific document and summary pairs. This model used a set of features to rank sentences for scientific paper summarization. Contractor et al. [22] proposed a model for extractive summarization to utilize the concept of argumentative zones (AZs) framework for academic papers. Kinugawa and Tsuruoka [23] proposed a two-level hierarchical structure based on encoder-decoder for extractive summarization of research papers.

But a common trend nowadays to supplement research highlights with full research paper and abstract. Hence, Highlights generation is another categorization task of text summarization. Collins et al. [24] proposed a supervised extarctive model for identifying a sentence is highlights or not also published urls for computer science publications as a benchmark dataset named CSPubSum. Alambo et al. [25] proposed a techniques for selecting salient language units and producing text in order to produce an abstractive summary of a scientific paper. L. Cagliero et al. [26] proposed an extractive approach based on gradient boosting method to select some sentences as a research highligths. Rehman at el. [27] proposed an abstractive method to generate research highlights from a research paper's abstract, by combining a pointer-generator model with Glove embeddings.

Our work is significantly different from the above works. Here we use pretrained ELMo embeddings with a pointer-generator model with coverage mechanism and generate research highlights using various sections of the paper. Generic texts are used to train the pretrained models like PEGASUS [28], T5 [29], (GPT(Generative

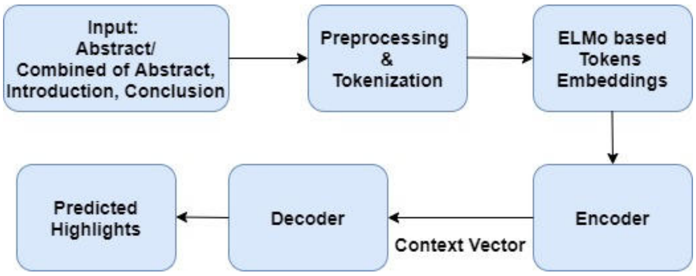

Fig. 3.1: Workflow used in our encoder-decoder model.

Pre-trained Transformer)-like) decoder [30], and BART [31] which appear to be the most effective summarizer. In order to have even better performance, they need to be fine-tuned using domain-specific inputs. However, this process needs significant resources. Our proposal describes a cost effective but useful architecture to meet the same objective.

3. Methodology. In this section, we describe the model we use to generate research highlights from the research articles. The workflow of our system is shown in Figure 3.1.

We conduct experiments with four different variations: (1) Pointer-generator model proposed by [17], (2) Incorporating coverage mechanism (proposed in [32]) into the pointer-generator model (the combined model is also referred to in the same work [17]), (3) Pretrained ELMo embeddings [33] with pointer-generator model, and (4) Pretrained ELMo embeddings with pointer-generator model and coverage mechanism.

3.1. ELMo Pre-trained word representations. Pre-trained word representations [34] have a significant role in many neural language understanding models. Pre-trained word representations are a key component in many neural language understanding models. On the other hand representation of learning with high quality and accurate generation are very challenging. A deep contextualized word representation, ELMo [33] can be used to capture the complex behaviour of word syntax, semantics and linguistic context. ELMo employs vectors that was trained with a stacked bidirectional LSTM.

3.2. Pointer-generator model with ELMo embeddings and coverage mechanism. We use a pointer-generator model, proposed by See, Liu and Manning [17]. It consists of a seq2seq model with a BiLSTM encoder and an LSTM decoder with attention [5]. However, instead of using word embeddings trained from scratch or non-contextual embeddings like word2vec [35] or GloVe [36], we use context-sensitive embeddings that represent homonyms with different vectors, despite the fact that these words have the same spelling. In other words, word representations capture the fine differences in meaning that arise from the context in which the words are used. In our present work, we add a pretrained ELMo contextual embeddings [33] layer that generates embedding for each word of the input text. Instead of directly passing the individual token id to the encoder recurrent neural network, we feed the token embeddings prepared by pretrained ELMo [33] embeddings layer. This can improve the model's ability to generate hidden states because the input words embedding matrix is initialized with the pretrained word embeddings ELMo. The embeddings are fine-tuned during model training. The dimension of ELMo word embeddings used in our experiment is 1024. The decoder has a unique copying technique, which decides between *copying* a word from the source text by utilizing copying mechanism or *generating* new words from the vocabulary (built from the vocabulary of the whole training corpus and the current input document). The copying mechanism helps to deal with out-of-vocabulary (OOV) words. The generating mechanism, on the other hand, induces new words which indicate novel paraphrasing. The decoder strikes a balance between copying words and generating words using a hyperparameter, which probabilistically chooses between the two alternatives. However, the pointer-generator model sometimes generates the same words repetitively. To overcome this problem, we used the coverage mechanism of Tu et al.[32]. In essence, this model focuses on the preceding time steps of the decoder through attention so that attending to the same word in the input document again and again is penalized.

4. Experimental setup.

4.1. Datasets. We make use of the computer science publication benchmark dataset published by Collins et al.[24] named CSPubSum, which contains the URLs of $\sim$ 10K papers from ScienceDirect [1]. The following fields are typically present in the documents: title, abstract, author-written research highlights, authors-written keywords, introduction, related work, experiment, conclusion, and other major subsections that are part of the discourse structure of a research paper. For our experiments, we divided the dataset into training, validation, and test subsets (train, val, test) in proportion of 80 : 10 : 10. In 98% of the papers, the highlights are at least 1.5 times shorter than the abstract. Therefore, research highlights can be viewed as a summary of both the abstract and the paper.

4.2. Data pre-processing. Before inputting the dataset to the model, we did some basic pre-processing steps. We removed unintended symbols, letters, urls, HTML tags and special characters. Then we changed the dataset to lowercase. To conduct experiments, we arranged the dataset in several ways. In particular, we organized it as *(abstract, research highlights written by author)*, *(abstract $\bigoplus$ introduction $\bigoplus$ conclusion, research highlights written by author)*, where text concatenation is represented as'$\bigoplus$'. When considering only abstract as an input, we allowed a maximum of 400 tokens. In the case of combined inputs from abstract, introduction and conclusion sections, we allowed a maximum of 1500 tokens. From each section, we allowed up to 500 tokens. In all cases, the token count of model-generated research highlights was limited to 100 only.

4.3. Implementation details. All the models were trained on the GPU-supported Colab Pro+ environment. The pointer-generator network with ELMo embeddings used 1024 as the word embedding dimension and that without ELMo embeddings used 128 as the word embedding dimension. For all models, the maximum vocabulary size was restricted to 50K tokens. For all models, the dimension of RNN hidden states is 256. We chose maximum gradient norm of 1.2 for gradient clipping.

4.4. Evaluation metrics. To compare the performance of the various models, we used the following metrics: ROUGE [7], METEOR [8], BERTscore [9], and MoverScore [10]. We have used ROUGE [7] metric to measure the word overlap between the research highlights written by the authors (ARRHS) and those generated by model (MGRHS). The recall (R), precision (P) and F1-measure ($F1$) for ROUGE-N are calculated as follows:

$$R = \frac{Matched\ number\ of\ n/grams\ in\ (\mathtt{MGRHS}, \mathtt{ARRHS})}{Number\ of\ n/grams\ in\ \mathtt{ARRHS}} \tag{4.1}$$

$$P = \frac{Matched\ number\ of\ n/grams\ in\ (\mathtt{MGRHS}, \mathtt{ARRHS})}{Number\ of\ n/grams\ in\ \mathtt{MGRHS}} \tag{4.2}$$

$$F1 = \frac{2 * (R * P)}{R + P} \tag{4.3}$$

A sequence of n words makes up an n-gram. ROUGE-L measures the longest common subsequence (LCS) between MGRHS and ARRHS. ROUGE-S measures the skip-bigram matched between MGRHS and ARRHS where a skip-bigram is a bigram that allows random word-gaps or skips between words. The recall based on skip-bigram is calculated as follows:

$$R_{skip} = \frac{Matched\ number\ of\ Skip/bigrams\ in\ (\mathtt{MGRHS}, \mathtt{ARRHS})}{Number\ of\ Skip/bigrams\ in\ \mathtt{ARRHS}} \tag{4.4}$$

The precision based on skip-bigram is calculated as follows:

$$P_{skip} = \frac{Matched\ number\ of\ Skip/bigrams\ in\ (\mathtt{MGRHS}, \mathtt{ARRHS})}{Number\ of\ Skip/bigrams\ in\ \mathtt{MGRHS}} \tag{4.5}$$

[1] (https://www.sciencedirect.com)

The F1-measure ($F1_{skip}$) based on skip-bigram is calculated as follows:

$$F1_{skip} = \frac{2 * (R_{skip} * P_{skip})}{R_{skip} + P_{skip}} \tag{4.6}$$

ROUGE-SU is an extension of ROUGE-S which counts both skip-bigram and unigram between MGRHS and ARRHS. According to the official ROUGE script, all of our ROUGE scores have a 95% confidence interval of at most ±0.25.

METEOR-score is calculated using an explicit word-to-word correspondence of research highlights generated by the model (MGRHS) and research highlights written by the authors (ARRHS).

BERTScore is calculated based on pairwise cosine similarity of each token in the research highlights generated by model (MGRHS) with that in the highlights written by authors (ARRHS). Here, instead of using the tokens directly, the similarity is computed based on contextual embeddings. When we tokenize the research highlights written by authors (ARRHS) and pass the tokens through the embedding model (in our case, ELMo), we get a sequence of contextual embeddings denoted as $\vec{x} = \langle \vec{x}_1, \ldots, \vec{x}_m \rangle$. Similarly, when we tokenize the research highlights generated by the model (MGRHS) and embed the tokens, we get a sequence of contextual embeddings denoted as $\hat{\vec{x}} = \langle \hat{\vec{x}}_1, \ldots, \hat{\vec{x}}_n \rangle$. The values of recall ($R_{\mathrm{BERT}}$), precision ($P_{\mathrm{BERT}}$), and F1-scores ($F_{\mathrm{BERT}}$) are computed as follows:

$$R_{\mathrm{BERT}} = \frac{1}{m} \sum_{\vec{x}_i \in \vec{x}} \max_{\hat{\vec{x}}_j \in \hat{\vec{x}}} \vec{x}_i^{\top} \hat{\vec{x}}_j \qquad P_{\mathrm{BERT}} = \frac{1}{n} \sum_{\hat{\vec{x}}_j \in \hat{\vec{x}}} \max_{\vec{x}_i \in \vec{x}} \vec{x}_i^{\top} \hat{\vec{x}}_j \tag{4.7}$$

$$F_{\mathrm{BERT}} = \frac{2 * (R_{\mathrm{BERT}} * P_{\mathrm{BERT}})}{R_{\mathrm{BERT}} + P_{\mathrm{BERT}}} \tag{4.8}$$

MoverScore [10] is calculated based on the contextualized representations and Word Mover's Distance (WMD) [37] between the research highlights generated by model (MGRHS) and the research highlights written by authors (ARRHS). It can take into account the presence of new or unseen words in the generated text, and evaluate how well they fit into the overall structure and content of the original text. It allows many-to-one alignment to map the semantically similar words in MGRHS and ARRHS whereas BERTScore considers only one-to-one alignment. The sentences of the research highlights written by the authors (ARRHS) and the research highlights generated by the model (MGRHS) are represented as x and $\hat{x}$. Their sequence of n-grams are denoted as x^n and $\hat{x}^n$. The transportation cost matrix (C) is calculated based on a distance metric (d) between the n-grams as follows:

$$C_{i,j} = d(x_i^n, \hat{x}_j^n) \tag{4.9}$$

where $d(x_i^n, \hat{x}_j^n)$ is the Euclidean distance between the i-th n-gram of x and the j-th n-gram of $\hat{x}$ where both the n-grams are represented by their respective embeddings. The authors in [10] define a transportation flow matrix F where $F(i,j)$ captures the amount of flow from the i-th n-gram (x_i^n) in x^n to the j-th n-gram ($\hat{x}_j^n$) in $\hat{x}^n$. Let $\langle C, F \rangle$ denote the sum of all elements in the matrix obtained from element-wise multiplication of C and F. We associate weights f_{x^n} and $f_{\hat{x}^n}$ with the n-grams x_n and $\hat{x}_n$, such that each n-gram gets a single weight value in each case and assume that each of f_{x^n} and $f_{\hat{x}^n}$ defines a probability distribution (i.e., the entries of each vector sums to 1). Finally, the moverscore [10] is defined as

$$\mathrm{WMD}(x^n, \hat{x}^n) = \min_{F \in \mathbf{R}^{|x^n| \times |\hat{x}^n|}} \langle C, F \rangle \qquad \text{such that} \;\; F\,\mathbf{1} = f_{x^n} \;\text{and}\; F^{\top}\mathbf{1} = f_{\hat{x}^n} \tag{4.10}$$

5. Results.

5.1. Comparison of four pointer-generator model variants. In this section, we compare various scores predicted by four variants of model with different types of input cases. The four variants of the model are (1) Pointer-generation model (**PGM**), (2) Pointer-generation model with coverage (**PGM + Cov**), (3)

Table 5.1: Pointer-generator type model evaluation: ROUGE, METEOR, BERTScore and MoverScore scores on different inputs from the CSPubSum dataset

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-S	ROUGE-SU	METEOR	BERTScore	MoverScore
abstract only	PGM	35.44	11.57	29.88	11.45	12.35	25.4	83.80	56.69
	PGM + Cov	36.57	12.3	30.69	12.14	13.04	25.4	84.05	57
	PGM + ELMo	35.74	12.54	32.47	11.81	12.71	19.75	82.62	54.98
	PGM + ELMo + Cov	**38.4**	**13.32**	**35.45**	**13.41**	**14.35**	**30.61**	**86.65**	**57.94**
abstract +introduction +conclusion	PGM	33.49	10.83	30.87	10.67	11.59	25.51	86.01	56.75
	PGM + Cov	35.73	11.61	32.96	11.6	12.52	27.71	86.26	57.39
	PGM + ELMo	33.6	11.44	30.97	10.78	11.69	25.68	86.02	56.79
	PGM + ELMo + Cov	**36.34**	**12.11**	**33.77**	**11.98**	**12.97**	**27.78**	**86.68**	**57.63**

Table 5.2: Comparison of the performance of the proposed model with that of other approaches for CSPubSum data set.

Model Name	ROUGE-2 (F1)	ROUGE-L (F1)
LSTM Classification [24]	12.7	29.50
Gradient Boosting Regressor [26]	**13.9**	31.60
Pointer-generator+ Coverage + GloVe [27]	8.57	29.14
PGM + ELMo + Coverage	13.32	**35.45**

Pointer-generation model with ELMo embeddings (**PGM + ELMo**), (4) Pointer-generation model with ELMo embeddings and coverage mechanism (**PGM + ELMo + Cov**). In models (1) and (2), the input contains word embeddings that are randomly initialized and trained with the model. Since we have proposed model (4) in this paper, investigation of the other three variants may be seen as an ablation study. For each model, the input could be the abstract only or a combination of abstract, introduction and conclusion of the paper.

Input: Abstract only:

When the input is a research paper's abstract, the results for ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-S, ROUGE-SU, METEOR, BERTScore and MoverScore are shown in Table 5.1. The pointer-generator model with ELMo embeddings and coverage mechanism achieves the best result in all cases.

Input: Abstract $\bigoplus$ Introduction $\bigoplus$ Conclusion:

When the input is a combination of abstract $\bigoplus$ introduction $\bigoplus$ conclusion, the results for ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-S, ROUGE-SU, METEOR, BERTScore and MoverScore are shown in Table 5.1 The pointer-generator model with ELMo embeddings and coverage mechanism displays the highest performance in all cases.

5.2. Comparison with previous works. On the CSPubSum dataset, as shown in the Table 5.2, we compare the performance of our proposed work with other prior works. We notice that our model, pointer-generator with ELMo embeddings and coverage mechanism (**PGM + ELMo + Cov**) achieves higher ROUGE-L scores than other methods in the literature.

6. Case studies. In this section, we have shown a few research highlights generated by our models to enable a qualitative study of the performance of the models. In all case studies, yellow color represents factual errors , orange color represents repeating phrases and green color identifies some correctly added words or phrases .

Figure 6.1 shows the comparison of predicted research highlights generated by variants of pointer-generator models when the input (for training and test) is the abstract of a research paper. Observe that the first model **PGM** generates repeating phrase 'total route duration', which is solved when we add coverage mechanism. Similarly, a full sentence is repeated by the **PGM + ELMo** model which the fourth model **PGM + ELMo + Cov** corrects using the coverage mechanism. Note that the words 'subtour elimination constraints' that the last model (**PGM + ELMo + Cov**) generates is present in the abstract of the paper, and its insertion in the output is semantically correct, although it is absent in the golden set of research highlights submitted by the

Abstract: “This paper introduces a version of the classical traveling salesman problem with time-dependent service times. In our setting, the duration required to provide service to any customer is not fixed but defined as a function of the time at which service starts at that location. The objective is to minimize the total route duration, which consists of the total travel time plus the total service time. The proposed model can handle several types of service time functions, e.g., linear and quadratic functions. We describe basic properties for certain classes of service time functions, followed by the computation of valid lower and upper bounds. We apply several classes of subtour elimination constraints and measure their effect on the performance of our model. Numerical results obtained by implementing different linear and quadratic service time functions on several test instances are presented.”

Author-written research highlights: “We introduce and model the traveling salesman problem with time dependent service times. The proposed model can handle several types of service time functions such as linear and quadratic. We describe properties of the service time function and compute valid lower and upper bounds. We propose and compare several mathematical programming formulations for the problem. Comparative computational results indicate that a flow based formulation is superior the other three.”

PGM: “We study minimize the total route duration of the total route duration . Consider the total route duration of the total time [UNK] describe basic properties for certain classes of service time functions. Model can handle several types of service time functions e.g. linear and quadratic functions. Examples obtained by valid lower and upper bounds are presented. Results obtained by implementing different quadratic service time functions are discussed.”

PGM + Cov: “We study the route duration of the total route duration of total service time. New model is based on various linear and valid lower heuristics. Results are conducted to illustrate different linear and quadratic service time functions. Results obtained by implementing linear and quadratic functions are presented.”

PGM + ELMo: “A version of the classical traveling salesman problem with time dependent service times is proposed . The proposed model can handle several types of service time functions e.g. linear and quadratic functions. The proposed model can handle several types of service time functions e.g. linear and quadratic functions. Numerical results obtained by implementing different linear service time functions are presented.”

PGM + ELMo + Cov: “This paper introduces a version of the classical traveling salesman problem with time dependent service times. The proposed model can minimize the total route duration which consists of the total service time. We apply several classes of subtour elimination constraints and measure their effect on the performance of our model . Numerical results obtained by implementing different test instances are presented.”

Fig. 6.1: The input consists of only the abstract of a paper from the CSPubSum dataset. The highlights generated by each of the four models are presented. The input abstract and the author-written research highlights are taken from `https://www.sciencedirect.com/science/article/pii/S037722171500702X`

authors. Another observation is that the models with **ELMo** embeddings display better linguistic quality with respect to grammatical syntax probably due to the contextual nature of the embeddings. For example, while the first model (**PGM**) contains a grammatically incorrect sentence like ‘We study minimize the ...’ and the second model (**PGM + Cov**) generates the incorrect sentence ‘Results are conducted ...’, the ELMo-based models do not display such issues. However, none of the models capture the last line “Comparative computational results indicate that a flow based formulation is superior the other three” of the highlights penned by the authors because it does not appear in the abstract.

Figure 6.2 depicts a similar comparison among the outputs of the four models for a different paper. Again, we notice that without the coverage mechanism, words are incorrectly repeated, while the coverage mechanism reduces repetition significantly. Grammatical correctness of ELMo-based models is also more than that of other models. Figure 6.3 shows the comparison of predicted research highlights generated by the models for the same paper when the input (for training and test) is the combination of a research paper’s abstract, introduction, and conclusion. Observe the same phenomenon of repetitive words in absence of the coverage mechanism: **PGM + ELMo** repeats the word ‘graphical’ several times, which is fixed when the coverage mechanism is added. However, a careful comparison of the outputs of **PGM + ELMo + Cov** across Figures 6.2 and 6.3 show that the highlights produced in the first case are more meaningful. For example, the statement ‘We modeled the image as a superpixel based markov random field segmentation framework’ (in **PGM + ELMo + Cov** model in 6.3) is not semantically correct. It seems that using a larger input from the body of the

Abstract: "Accurate and effective cervical smear image segmentation is required for automated cervical cell analysis systems. Thus, we proposed a novel superpixel-based Markov random field (MRF) segmentation framework to acquire the nucleus, cytoplasm and image background of cell images. We seek to classify color non-overlapping superpixel-patches on one image for image segmentation. This model describes the whole image as an undirected probabilistic graphical model and was developed using an automatic label-map mechanism for determining nuclear, cytoplasmic and background regions. A gap-search algorithm was designed to enhance the model efficiency. Data show that the algorithms of our framework provide better accuracy for both real-world and the public Herlev datasets. Furthermore, the proposed gap-search algorithm of this model is much more faster than pixel-based and superpixel-based algorithms."

Author-written research highlights: "We proposed a novel gap search markov random field mrf for accurate cervical smear image segmentation. This method could acquire three regions nuclei cytoplasm and background automatically by a label map mechanism. The gap search algorithm is faster than other three algorithms in the experiments. A copy of source codes will be released as an open source project for continuing studies."

PGM: "A novel model is developed to classify color non overlapping superpixel patches . Segmentation is used for the first time for image segmentation . Gap search algorithm for one image segmentation and superpixel based algorithms. Results are much more faster than pixel based and superpixel based model. Algorithms are more faster than pixel of superpixel based algorithms."

PGM + Cov: " We seek to classify color non overlapping superpixel patches on one image image . Work presents an label map mechanism to acquire the whole image as the undirected probabilistic graphical model . Gap search algorithm was designed to enhance the model efficiency. Algorithms much more faster than pixel based and superpixel superpixel ."

PGM + ELMo: "We propose a novel superpixel based markov random field mrf segmentation framework . The model describes the nucleus cytoplasm and image background of cell images. A gap search algorithm is designed to solve the model efficiency. The gap search algorithm is much more faster than pixel based and superpixel based algorithms."

PGM + ELMo + Cov: "A novel image segmentation method for automated cervical cell analysis is proposed. The whole image as an undirected probabilistic graphical model is presented . A gap search method is proposed to solve the nuclear cytoplasmic and background regions . The proposed algorithm is much more faster than pixel based and superpixel based algorithms."

Fig. 6.2: The input is only the abstract of a paper from the CSPubSum dataset. The highlights generated by each of the four models are presented. The input paper is at `https://www.sciencedirect.com/science/article/pii/S0010482516300154`

paper, rather than only the abstract, confuses the model, resulting in lower-quality output. Recollect that the superior performance of abstract-only models is quantitatively communicated by Table 5.1, and we now see a specific qualitative example of the performance difference. None of the models, irrespective of the input type, includes the information in the sentence "A copy of source codes will be released as an open source project for continuing studies." mentioned as the last line of the author-written highlights.

Thus, in all cases, we observe that the research highlights generated by the pointer-generator with ELMo embeddings and the coverage mechanism with the abstract as input are semantically more meaningful and contains fewer syntactic issues compared to the outputs in other cases. However, we also note that the research highlights produced by the models do not agree perfectly with the author-written highlights when the latter contain information not included in the inputs to the model.

7. Conclusion and future works. We analyzed four variants of deep neural models with two different input cases on CSPubSum dataset for the task of research highlight generation. The best performance was obtained by the pointer-generator model that utilized ELMo embeddings and coverage mechanism. Our future aim is to delve into the reasons behind the enhancement of model performance with pretrained embeddings. We also intend to integrate additional techniques to enhance the quality of the generated research highlights, both semantically and syntactically.

Author-written research highlights: "We proposed a novel gap search markov random field mrf for accurate cervical smear image segmentation. This method could acquire three regions nuclei cytoplasm and background automatically by a label map mechanism. The gap search algorithm is faster than other three algorithms in the experiments. A copy of source codes will be released as an open source project for continuing studies."

PGM: "We propose a novel superpixel based markov segmentation model . we combined the superpixels with an undirected probabilistic model. we combined the superpixels with an automatic label map mechanism. we combined the superpixels with an automatic label map mechanism. "

PGM + Cov: A novel superpixel based markov random field segmentation model to acquire the nucleus cytoplasm and image background of cell images we combined the superpixels with an undirected probabilistic graphical model and connecting edges . First the connecting edges of the graph reflect spatial information between neighboring superpixels by local probabilistic. second the connecting edges of the graph reflect spatial information between neighboring superpixels by local ."

PGM + ELMo: "A probabilistic graphical graphical graphical graphical graphical graphical graphical graphical graphical graphical graphical graphical graphical graphical an automatic superpixel based mrf segmentation model is proposed to acquire the requisite accuracy of the proposed gap of the proposed gap a superpixel based mrf segmentation model is proposed to enhance the model efficiency. The proposed framework is developed using an automatic label map mechanism for determining nuclear cytoplasmic and background regions."

PGM + ELMo + Cov: "We proposed a probabilistic graphical framework to acquire the nucleus cytoplasm and background in cervical smear images . We modeled the image as a superpixel based markov random field segmentation framework. A gap search algorithm was developed using an automatic label map mechanism for determining the model efficiency."

Fig. 6.3: The input consists of the concatenation of the abstract, introduction, and conclusion of a paper from the CSPubSum dataset. The highlights generated by each of the four models are presented. The input paper is at `https://www.sciencedirect.com/science/article/pii/S0010482516300154`

REFERENCES

[1] Richard Van Noorden. Global scientific output doubles every nine years. *Nature news blog*, 2014.

[2] Lutz Bornmann, Robin Haunschild, and Rüdiger Mutz. Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases. *Humanities and Social Sciences Communications*, 8(1):1–15, 2021.

[3] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[4] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 68–73, 1995.

[5] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 280–290, Berlin, Germany, 2016.

[6] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT. *arXiv:2302.09419*, 2023.

[7] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81, 2004.

[8] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72, 2005.

[9] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, (ICLR 2020)*, 1–43, 2020.

[10] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *EMNLP-IJCNLP*, 563–578. Association for Computational Linguistics, November 2019.

[11] Phyllis B Baxendale. Machine-made index for technical literature—an experiment. *IBM Journal of research and development*, 2(4):354–361, 1958.

[12] Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[13] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. 2010.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014)*, 3104–3112, 2014.

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 1–15, 2015.

[16] Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 93–98, 2016.

[17] A. See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol.1, 1073–1083, 2017.

[18] Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. *preprint arXiv:1808.10792*, 2018.

[19] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

[20] Dang Trung Anh and Nguyen Thi Thu Trang. Abstractive text summarization using pointer-generator networks with pre-trained word embedding. In *Proceedings of the 10th International Symposium on Information and Communication Technology*, 473–478, 2019.

[21] Nouf Ibrahim Altmami and Mohamed El Bachir Menai. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[22] Danish Contractor, Yufan Guo, and Anna Korhonen. Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012*, 663–678, December 2012.

[23] Kazutaka Kinugawa and Yoshimasa Tsuruoka. A hierarchical neural extractive summarizer for academic papers. In *New Frontiers in Artificial Intelligence: JSAI-isAI Workshops, JURISIN, SKL, AI-Biz, LENLS, AAA, SCIDOCA, kNeXI, Tsukuba, Tokyo, November 13-15, 2017, Revised Selected Papers 9*, 339–354. Springer, 2018.

[24] Ed Collins, Isabelle Augenstein, and Sebastian Riedel. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada, August 2017.

[25] Amanuel Alambo, Cori Lohstroh, Erik Madaus, Swati Padhee, Brandy Foster, Tanvi Banerjee, Krishnaprasad Thirunarayan, and Michael Raymer. Topic-centric unsupervised multi-document summarization of scientific and news articles. In *2020 IEEE International Conference on Big Data (Big Data)*, 591–596, 2020.

[26] Luca Cagliero and Moreno La Quatra. Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications*, 160:113659, 2020.

[27] Tohida Rehman, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. Automatic generation of research highlights from scientific. In *2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021), collocated with JCDL 2021*, 2021.

[28] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Procs. of the International Conference on Machine Learning*, 11328–11339. PMLR, 2020.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *preprint arXiv:1910.10683*, 2019.

[30] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *preprint arXiv:1910.13461*, 2019.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[32] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 76–85, August 2016.

[33] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2227–2237, June 2018.

[34] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543, October 2014.

[35] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

[36] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543, 2014.

[37] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, 957–966. PMLR, 2015.

Edited by: Katarzyna Wasielewska
Received: Apr 12, 2023
Accepted: Jun 8, 2023