

Named Entity Recognition Based Automatic Generation of Research Highlights

Tohida Rehman

Jadavpur University, India Indian Association for the Cultivation of Science
tohida.rehman@gmail.com debarshisanyal@gmail.com

Debarshi Kumar Sanyal

Prasenjit Majumder

TCG CREST, India
prasenjit.majumder@gmail.com

Samiran Chattopadhyay

TCG CREST; Jadavpur University, India
samirancju@gmail.com

Abstract

A scientific paper is traditionally prefaced by an abstract that summarizes the paper. Recently, research highlights that focus on the main findings of the paper have emerged as a complementary summary in addition to an abstract. However, highlights are not yet as common as abstracts, and are absent in many papers. In this paper, we aim to automatically generate research highlights using different sections of a research paper as input. We investigate whether the use of named entity recognition on the input improves the quality of the generated highlights. In particular, we have used two deep learning-based models: the first is a pointer-generator network, and the second augments the first model with coverage mechanism. We then augment each of the above models with named entity recognition features. The proposed method can be used to produce highlights for papers with missing highlights. Our experiments show that adding named entity information improves the performance of the deep learning-based summarizers in terms of ROUGE, METEOR and BERTScore measures.

from the original text. Scientific papers are generally longer documents than news stories and have a different discourse structure. Additionally, there are less resources available on scholarly documents to train text summarization systems. There are two broad approaches used in automatic text summarization (Luhn, 1958; Radev et al., 2002): Extractive summarization and abstractive summarization. Extractive summarization generally copies whole sentences from the input source text and combines them into a summary, discarding irrelevant sentences from the input (Jing and McKeown, 2000; Knight and Marcu, 2002). But recent trends use abstractive summarization which involves natural language generation to produce novel words and capture the salient information from the input text (Rush et al., 2015; Nallapati et al., 2016). Our aim is to generate research highlights from a research paper using an abstractive approach. But an abstractive summarizer using a generative model like a pointer-generator network (See et al., 2017) sometimes generates meaningless words in the output. In particular, for named entities which are multi-word strings, incorrect generation of a single word within the string (e.g., suppose instead of generating ‘artificial neural network’, it generates ‘artificial network’) may corrupt the meaning of the whole entity and its parent sentence. So we propose to perform named entity recognition (NER) on the input and treat a named entity as a single token before the input passes through the summarizer. This will avoid their fragmentation in the output.

The main contributions of this paper are:

1. We propose a mechanism to combine named entity recognition with pointer-generator networks having coverage mechanism to automatically generate research highlights, given the abstract of a research paper. To the best of our knowledge, this work is the first attempt to use NER in pointer-generators with coverage mechanism (See et al., 2017) to generate

1 Introduction

Every research domain has an overabundance of textual information, with new research articles published on a daily basis. The number of scientific papers is increasing at an exponential rate (Bornmann et al., 2021). According to reports, the number of scientific articles roughly doubles every nine years (Van Noorden, 2014). For a researcher, keeping track of any research field is extremely difficult even in a narrow sub-field. Nowadays, many publishers request authors to provide a bulleted list of research highlights along with the abstract and the full text. It can help the reader to quickly grasp the main contributions of the paper.

Automatic text summarization is a process of shortening a document by creating a gist of it. It encapsulates the most important or relevant information

research highlights.

2. We analyze the performance of generating research highlights for the following different input types: (a) the input is the abstract only, (b) the input comprises the abstract and the conclusion, (c) the input comprises the introduction and the conclusion.
3. We evaluate the performance of the models using ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BERTScore (Zhang et al., 2020b) metrics.

2 Literature survey

Early works on summarization of scientific articles include (Kupiec et al., 1995) where an extractive summarization technique is proposed and evaluated on a small dataset of 188 scientific documents, and (Teufel and Moens, 2002) which exploits the rhetorical status of assertions to summarize scientific articles. More recently, Lloret et al. (Lloret et al., 2013) have developed a new corpus of computer science papers from arXiv.org that contains pairs of (Introduction, Abstract). An approach proposed to generate abstracts from a research paper using Multiple Timescale model of the Gated Recurrent Unit (MTGRU) may be found in (Kim et al., 2016). Surveys on summarization of scholarly documents appears in (Altmami and Menai, 2020), (El-Kassas et al., 2021). Generating research highlights from scientific articles is not the same as document summarization. A supervised machine learning approach is proposed in (Collins et al., 2017) to identify relevant highlights from the full-text of a paper using a binary classifier. They have also contributed a new benchmark dataset containing author written research highlights for more than 10,000 documents. All documents adhere to a consistent discourse structure. Instead of a simple binary classification of sentences as highlights or not, (Cagliero and La Quatra, 2020) used multivariate regression methods to select the top- K most relevant sentences as research highlights. (Hassel, 2003) proposed a method to use appropriate weight for the named entity tagger into the SweSum summarizer for Swedish newspaper texts. (Marek et al., 2021) proposed an extractive summarization technique that determines a sentence’s significance based on the density of named entities. (Rehman et al., 2021) used a pointer-generator model with coverage (See et al., 2017) to gener-

ate research highlights from abstracts. The present work, unlike the existing ones, uses NER to avoid incorrect phrases from being generated by the decoder. Note that pretrained summarization models like PEGASUS (Zhang et al., 2020a), T5 (Raffel et al., 2019), and BART (Devlin et al., 2019) are trained on generic texts. Fine-tuning them to a specific (e.g., scientific) domain requires large memory and computational resources; in this context, this paper provides a simpler alternative.

3 Methodology

We use a pointer-generator network (See et al., 2017) as our baseline model. While the pointer-generator model (See et al., 2017) first tokenizes a document using Stanford CoreNLP tokenizer and converts the tokens to word embeddings (trained with the model), the method we propose here performs NER on the input document and considers a named entity as a single token when training the model. We perform experiments with 4 variants: (1) the original pointer-generator model proposed in (See et al., 2017) (PGM), (2) pointer-generator model integrating coverage mechanism (proposed in (Tu et al., 2016)) (PGM + Cov), described in the same work (See et al., 2017), (3) NER-based pointer-generator model (NER + PGM), and (4) NER-based pointer-generator model with coverage mechanism (NER + PGM + Cov).

3.1 NER-based pointer-generator network

This model consists of an NER-based tokenizer layer and a pointer-generator network. The NER-based tokenizer layer converts the words in the input document to a sequence of tokens, thus preserving an entity name as a single token. In particular, it uses the named entity recognizer in spaCy¹ However, we do not use entity types. We do not use pretrained word embeddings as (Nallapati et al., 2016) do; in our case token embeddings are learned from scratch during training. Here, the main role of NER is that instead of directly feeding the normal tokens of the input document into the encoder, we are passing the NER-based tokens.

4 Experimental setup

4.1 Data sets

We use the data sets published by Collins et al. (Collins et al., 2017), which

¹<https://spacy.io/usage/linguistic-features>.

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
abstract only	PGM	35.44	11.57	29.88	25.4	83.80
	PGM + Cov	36.57	12.3	30.69	25.4	84.05
	NER + PGM	35.88	12.78	33.21	29.14	86.02
	NER + PGM + Cov	38.13	13.68	35.11	31.03	86.3
abstract + conclusion	PGM	29.85	8.16	25.80	19.38	83.19
	PGM + Cov	31.70	8.31	26.72	20.92	83.49
	NER + PGM	35.12	12.37	32.37	28.34	86.08
	NER + PGM + Cov	37.48	13.26	34.95	28.97	86.64
introduction + conclusion	PGM	29.78	7.47	25.15	19.25	83.05
	PGM + Cov	31.63	7.65	26.25	20.24	83.32
	NER + PGM	31.74	9.18	29.44	23.82	85.78
	NER + PGM + Cov	34.24	9.82	31.92	25.36	86.1

Table 1: Evaluation of pointer-generator type models: F1-scores for ROUGE, METEOR and BERTScore on various inputs from CS PubSumm dataset. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script.

contains URLs of 10147 computer science publications from ScienceDirect (<https://www.sciencedirect.com/>). Title, abstract, author-written research highlights, a list of keywords referenced by the authors, introduction, related work, experiment, conclusion, and other important subsections as found in typical research papers are all included for each document. In our setup, every example in this data set is organised as follows: (*abstract, author-written research highlights, introduction, and conclusion*). We use 8116 examples for training, 1017 examples for validation, and 1014 examples for testing.

4.2 Data processing

We have removed digits, punctuation, and special characters from the documents and lowercased the entire corpus. The `retokenizer.merge` method of spaCy is used to tokenize and merge several tokens into one single token based on the named entities in the document. Instead of individual tokens of “artificial”, “neural”, and “network”, we pass all the three tokens together as a single token “artificial neural network” (referenced as vocab index 17). The data set is then reorganized in several ways to conduct various experiments. We organize the data set as (*abstract, author-written research highlights*), (*abstract + conclusion, author-written research highlights*), and (*introduction + conclusion, author-written research highlights*) where ‘+’ denotes text concatenation. Since abstract and introduction usually emphasize the same aspects of the paper, we have not included them together. In this data set, the average abstract length is 186 tokens, while the average

author-written-highlight length is 52 tokens. When we considered abstract and conclusion, the average length was 643. When we considered introduction and conclusion, the average length was 1234. Therefore in our model, we have set the maximum number of input tokens to 400 when the abstract is taken as the input. For all other cases, the maximum count of input tokens is set to 1500. In all cases, the generated research highlights have a maximum token count of 100. We trained all models on Tesla V100-SXM2-16GB Colab Pro+ that supports GPU-based training. For all models, we used two bidirectional LSTMs with cell size of 256, word embeddings of dimension 128, and maximum vocabulary size of 50K tokens. We considered gradient clipping with a maximum gradient norm of 1.2. We use other hyperparameters as suggested by (See et al., 2017).

4.3 Comparison with previous works

Table 2 compares the performance of our proposed approach (NER + PGM + Cov) with other competitive baselines in the literature, on the CS PubSumm data set in terms of the ROUGE-L F1-score.

Model Name	ROUGE-L (F1)
LSTM (Collins et al., 2017)	29.50
GBR (Cagliero and La Quatra, 2020)	31.60
PGM + Cov + GloVe (Rehman et al., 2021)	29.14
NER + PGM + Cov (ours)	35.11

Table 2: Comparison of ROUGE-L (F1-score) of the proposed model with that of the other approaches.

Author-written research highlights: This paper studies an image collection planning problem for a korean satellite kompsat 2(korea multi purpose satellite 2). We present a binary integer programming model for this problem in a multi orbit long term planning environment. A heuristic solution approach based on the lagrangian relaxation and subgradient methods is provided.
PGM: Image collection with 2 korea 2 image acquisition [UNK] of segment square lengths as well as well as collection relaxation . Integer programming model exploiting lagrangian relaxation. Results of our computational experiment based on the lagrangian relaxation and subgradient methods.
PGM + Cov: Image collection planning problem with a multi orbit multi purpose satellite 2 korea multi purpose. A binary integer programming model for a multi orbit long term planning. A heuristic solution approach based on the lagrangian relaxation and subgradient methods. A heuristic solution approach based on the lagrangian relaxation and subgradient methods .
NER + PGM: We present a binary integer programming model for this image collection planning problem for a korean satellite kompsat. We present a binary integer programming model for this problem in a multi orbit long term planning environment. Combining the lagrangian relaxation and subgradient methods using the lagrangian relaxation and subgradient methods to solve the top problem.
NER + PGM + Cov: We present an image collection planning problem for a korean satellite kompsat. We present a binary integer programming model for image collection planning. We show the heuristic approach based on the lagrangian relaxation. We present the results on a multi orbit long term planning environment.

Figure 1: Input is only an abstract from CSPubSumm data set. Highlights produced by each of the four models are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S037722171300307X>

5 Results

5.1 Comparison of pointer-generator type models

In this sub-section, we report the results of experiments on the CSPubSumm data set for various input types. Table 1 shows the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR and BERTScore metrics for various inputs from the test dataset. Among the four models, the NER-based pointer-generator network with coverage mechanism achieves the highest ROUGE, METEOR and BERTScore values. It appears that treating an entity as a single token in the input helps to learn better embeddings and results in more controlled generation of the output, thereby reducing semantically invalid words and phrases. We aim to investigate this aspect in future. The (NER + PGM + Cov) model achieves the highest scores when the input is the abstract, indicating that most of the findings reported by the research highlights are already in the abstract, and adding additional sections to the input contributes to noise for the model.

5.2 Case study

Figure 1, 2 and 3 shows sample outputs generated by our models for various input types. In all the case studies reported below, we highlight examples of factual errors , repeating words and correct named entities . Note that the NER-based models correctly generate the named entities in the output.

5.3 Manual evaluation

We selected a set of 25 papers, their author-written highlights (**A**) and their highlights from only the (NER + PGM + Cov) model (**M**). We recruited 16 human annotators (possessing or pursuing advanced degrees in software engineering at premier universities in India) to independently rate a given summary on a scale of 1(low) to 5(high) for adequacy and fluency (separately). Each rater was given the full text of a paper and *either* the author-written *or* the machine-generated highlights of the paper, but not told which one. Each summary was rated independently by two raters. On fluency, the average score for **A** was 4.02 and that for **M** was 3.3, while on adequacy, the average score for **A** was 3.82 and that for **M** was 3.12. This shows the machine-generated highlights are only slightly worse than the human-written ones.

6 Conclusion

We applied four different deep neural models to generate research highlights from a research paper. The NER-based pointer-generator model with coverage mechanism achieved the best performance in all cases. In future, we intend to investigate in detail why NER improves the generated highlights. We also aim to ensure that the generated highlights are correct in syntax and semantics.

<p>Author-written research highlights: We present a lightweight non parametric approach to generate wrinkles for 3d facial modeling and animation. Our method represents a convenient approach for generating plausible facial wrinkles with low cost. Our method enables the reconstruction of captured expressions with wrinkles in real time.</p>
<p>PGM: We propose a non parametric facial modeling 3d face models from the 3d facial modeling . Synthesize the 3d face expression model with 3d depth camera is considered. Propose a non parametric face method to evaluate the performance of the 3d facial modeling . Method is provided to evaluate the performance of the proposed approach with respect to the existing method.</p>
<p>PGM + Cov: We propose non parametric face acquisition 3d facial modeling models. Face expression model is based on the texture synthesis of multiple subjects. Synthesis method guarantees to 3d face face acquisition. One high quality 3d face model is studied.</p>
<p>NER + PGM: A novel synthesis method is proposed to enhance the wrinkles using a single low cost kinect camera. The lightweight feature of the method is that it can generate plausible wrinkles using a single low cost kinect camera and one high quality 3d face model with details as the example. User specific expressions are used as blendshapes to capture facial animations in real time kinect camera and one high quality 3d face model with details.</p>
<p>NER + PGM + Cov: We present a lightweight non parametric method to generate wrinkles for 3d facial modeling and animation. The lightweight feature of the method is that it can generate plausible wrinkles. Our method is low cost and convenient for common users.</p>

Figure 2: Input is (abstract + conclusion) from CSPubSumm data set. Highlights produced by each of the four models are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S0010448514001857>

<p>Author-written research highlights: We propose a novel parallel 3d delaunay meshing algorithm for large scale simulations. The model information is kept during parallel triangulation process. A 3d local non delaunay mesh repair algorithm is proposed. The meshing results can be very approaching to the model boundary. The method can achieve high parallel performance and perfect scalability.</p>
<p>PGM: We propose a solid model boundary preserving method for large scale parallel 3d delaunay meshing. Parallel 3d local mesh 3d delaunay meshing algorithm is proposed. Mesh reconstruction is iteratively performed to meet both the mesh and the shared interfaces . Propose a parallel 3d local mesh reconstruction algorithm to construct delaunay triangulation . Results show high performance and perfect scalability.</p>
<p>PGM + Cov: A new semantic parallel algorithm is proposed for large scale parallel 3d delaunay meshing. Numerical local mesh is the sampling vertices for the problem 3d delaunay meshing. Propose a parallel su based partitioning algorithm by solving the algorithm . Proposed algorithm is highly parallelized to large scale sets and high quality partition walls.</p>
<p>NER + PGM: We propose a solid model boundary preserving method for large scale 3d delaunay meshing. The 3d boundary representation model information is kept during the entire parallel 3d delaunay triangulation process. The 3d boundary representation model information is kept during the entire parallel 3d delaunay triangulation process . A parallel 3d local mesh optimization algorithm is presented. Experimental results demonstrate high performance and perfect scalability.</p>
<p>NER + PGM + Cov: We propose a solid model boundary preserving method for large scale parallel delaunay meshing. The 3d boundary representation model information is during the entire parallel 3d delaunay triangulation process. A parallel local mesh refinement algorithm to repair the non delaunay mesh is proposed. A parallel 3d delaunay mesh refinement is presented. Experimental results demonstrate scalability performance.</p>

Figure 3: Input is (introduction + conclusion) from CSPubSumm data set. Highlights produced by each of the four models are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S0010448514001821>

References

- Nouf Ibrahim Altmami and Mohamed El Bachir Menai. 2020. [Automatic summarization of scientific articles: A survey](#). *Journal of King Saud University-Computer and Information Sciences*.
- Satanjeev Banerjee and Alon Lavie. 2005. [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*, pages 65–72.
- Lutz Bornmann, Robin Haunschild, and Rüdiger Mutz. 2021. [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.
- Luca Cagliero and Moreno La Quatra. 2020. [Extracting highlights of scientific articles: A supervised summarization approach](#). *Expert Systems with Applications*, 160:113659.
- Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. [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. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [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)*, pages 4171–4186.
- Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. [Automatic text summarization: A comprehensive survey](#). *Expert Systems with Applications*, 165:113679.
- Martin Hassel. 2003. [Exploitation of named entities in automatic text summarization for swedish](#). In *NODALIDA'03-14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland, May 30-31 2003*, page 9.
- Hongyan Jing and Kathleen McKeown. 2000. [Cut and paste based text summarization](#). In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.
- Minsoo Kim, Dennis Singh Moirangthem, and Minhoo Lee. 2016. [Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization](#). In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 70–77, Berlin, Germany. Association for Computational Linguistics.
- Kevin Knight and Daniel Marcu. 2002. [Summarization beyond sentence extraction: A probabilistic approach to sentence compression](#). *Artificial Intelligence*, 139(1):91–107.
- Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. [A trainable document summarizer](#). In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81.
- Elena Lloret, María Teresa Romá-Ferri, and Manuel Palomar. 2013. [Compendium: A text summarization system for generating abstracts of research papers](#). *Data & Knowledge Engineering*, 88:164–175.
- Hans Peter Luhn. 1958. [The automatic creation of literature abstracts](#). *IBM Journal of research and development*, 2(2):159–165.
- Petr Marek, Štěpán Müller, Jakub Konrád, Petr Lorenc, Jan Pichl, and Jan Šedivý. 2021. [Text summarization of Czech news articles using named entities](#). *arXiv preprint arXiv:2104.10454*.
- Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. [Abstractive text summarization using sequence-to-sequence rnns and beyond](#). *arXiv preprint arXiv:1602.06023*.
- Dragomir Radev, Eduard Hovy, and Kathleen McKeown. 2002. [Introduction to the special issue on summarization](#). *Computational Linguistics*, 28(4):399–408.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *arXiv preprint arXiv:1910.10683*.
- Tohida Rehman, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. 2021. [Automatic generation of research highlights from scientific abstracts](#). In *EEKE@JCDL*, pages 69–70.
- Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). *arXiv preprint arXiv:1509.00685*.
- A. See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Simone Teufel and Marc Moens. 2002. [Articles summarizing scientific articles: Experiments with relevance and rhetorical status](#). *Computational Linguistics*, 28(4):409–445.

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

Richard Van Noorden. 2014. Global scientific output doubles every nine years. *Nature News blog*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020b. [BERTScore: Evaluating text generation with BERT](#). In *International Conference on Learning Representations*.