Human-Centred Natural Language Processing

Abstractive Text Summarization using Natural Language Processing

Chinu Mangal Debapriya Roy Harshit Kumar Tyagi Piyush Vikas
Shweta Bambal Vishal Rajesh Kushwaha

SUPERVISOR: Dr. Marco Polignano
Dr. Purificato
Prof. Deluca

Agenda

- ➔ Text Summarization
- ➔ Literature Review
- ➔ Project Objectives and Goals
- ➔ Dataset
- ➔ Preprocessing & Tokenization
- ➔ Model and Methodology
- ➔ Evaluation Metrics
- ➔ Results and Analysis
- ➔ Limitations
- ➔ Future Work
- ➔ Conclusion
- ➔ References

Introduction to Text Summarization

- Process in artificial intelligence where a computer program reduces a text document into a shorter version, preserving only the essential or most important information.
- Increasingly important as the amount of digital information is growing and people need to process information more efficiently.
- Crucial role in various applications like information retrieval, Document summarization, content understanding, etc.
- Extractive vs. Abstractive Summarization

Extractive Text Summarization

- Involves selecting and extracting important sentences or phrases directly from the source text.
- Algorithm ranks segments of texts based on features like frequency, position, and thematic words.

Abstractive Text Summarization

- Generating summaries that go beyond simple extraction of sentences. Involves understanding the meaning of the text and generating novel sentences to convey the essential information.
- Enables the summarizer to condense information and paraphrase content, differs from extractive summarization by potentially offering more coherent and fluent summaries, similar to humans.
- Challenges in Generating Summaries:
  - Understanding Context
  - Maintaining Coherence
  - Handling Ambiguity

Evolution of Text Summarization

The development of effective text summarization algorithms relies heavily on natural language processing (NLP) and machine learning. Techniques such as deep learning, particularly the use of neural networks like Recurrent Neural Networks (RNNs) and Transformers, are at the forefront of advanced summarization.

- Early approaches: Rule-based and statistical methods.
- Advancements: Introduction of machine learning techniques, such as deep learning.
- Current state: Transformer-based models dominate the field, achieving remarkable results in both extractive and abstractive summarization.

Literature Review

PEGASUS Model by Hugging Face

- PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization.
- Developed by Google, it stands for "Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence model". It is a state-of-the-art model specifically designed for abstractive text summarization.

Literature Review

Key Features of PEGASUS:

- Pre-training Technique: PEGASUS is distinct in its pre-training approach. Unlike BERT/GPT, which use random masking of words, PEGASUS masks whole sentences or important parts of the text which encourages the model to focus more on salient content, thereby enhancing its summarization capabilities.
- Self-Attention Mechanisms: Utilizes the Transformer architecture's self-attention mechanisms to better understand the context of the entire document, facilitating more coherent and contextually accurate summaries.

Literature Review

Performance and Applications:

- PEGASUS has been shown to better perform in terms of both speed and accuracy on benchmark datasets like CNN/DailyMail and XSum.
- It is widely used for news articles, scientific papers, and any domain requiring high-quality summary generation.

Implementation by Hugging Face:

- Hugging Face offers an accessible implementation of PEGASUS, making it easy to integrate and use within various applications.
- Their platform provides APIs that allow developers to leverage PEGASUS for their specific summarization needs, backed by a community that contributes to ongoing improvements and iterations of the model.

Project Objectives and Goals

Objectives:

- Develop an abstractive text summarization system using the PEGASUS model.
- Utilize the Samsum dataset from the HuggingFace API for training and evaluation.

Goals:

- Achieve high-quality summaries that capture the key information of the input text.
- Evaluate the performance of the PEGASUS model using standard evaluation metrics such as ROUGE.

Dataset

.

Dataset Viewer — Auto-converted to Parquet | API | View in Dataset Viewer

Split (3)
train · 14.7k rows

Search this dataset

id string · *lengths* 8 – 10	dialogue string · *lengths* 0 – 5.49k	summary string · *lengths* 3 – 300
13818513	Amanda: I baked cookies. Do you want some? Jerry: Sure! Amanda: I'll bring you tomorrow :-)	Amanda baked cookies and will bring Jerry some tomorrow.
13728867	Olivia: Who are you voting for in this election? Oliver: Liberals as always. Olivia: Me too!! Oliver:…	Olivia and Olivier are voting for liberals in this election.
13681000	Tim: Hi, what's up? Kim: Bad mood tbh, I was going to do lots of stuff but ended up procrastinating Tim:…	Kim may try the pomodoro technique recommended by Tim to get more stuff done.
13730747	Edward: Rachel, I think I'm in ove with Bella.. rachel: Dont say anything else.. Edward: What do you…	Edward thinks he is in love with Bella. Rachel wants Edward to open his door. Rachel is outside.
13728094	Sam: hey overheard rick say something Sam: i don't know what to do :-/ Naomi: what did he say?? Sam: he…	Sam is confused, because he overheard Rick complaining about him as a roommate. Naomi thinks Sam should talk…
13716343	Neville: Hi there, does anyone remember what date I got married on? Don: Are you serious? Neville: Dead…	Wyatt reminds Neville his wedding anniversary is on the 17th of September. Neville's wife is upset and it…

‹ Previous 1 2 3 ... 148 Next ›

Fig.1 SAMsun Dataset

Preprocessing and Tokenization

- Utilize ‘datasets’ library to load SAMsum dataset
- Split into train, test and validation sets
- Tokenization : ‘AutoTokenizer’ from transformers library; compatible with Pegasus model. Each dialogue and summary is tokenized to a sequence of token IDs
- Converted text data into format (input IDs, attention masks, labels) in batches, preparing for training with the model.

Model Architecture

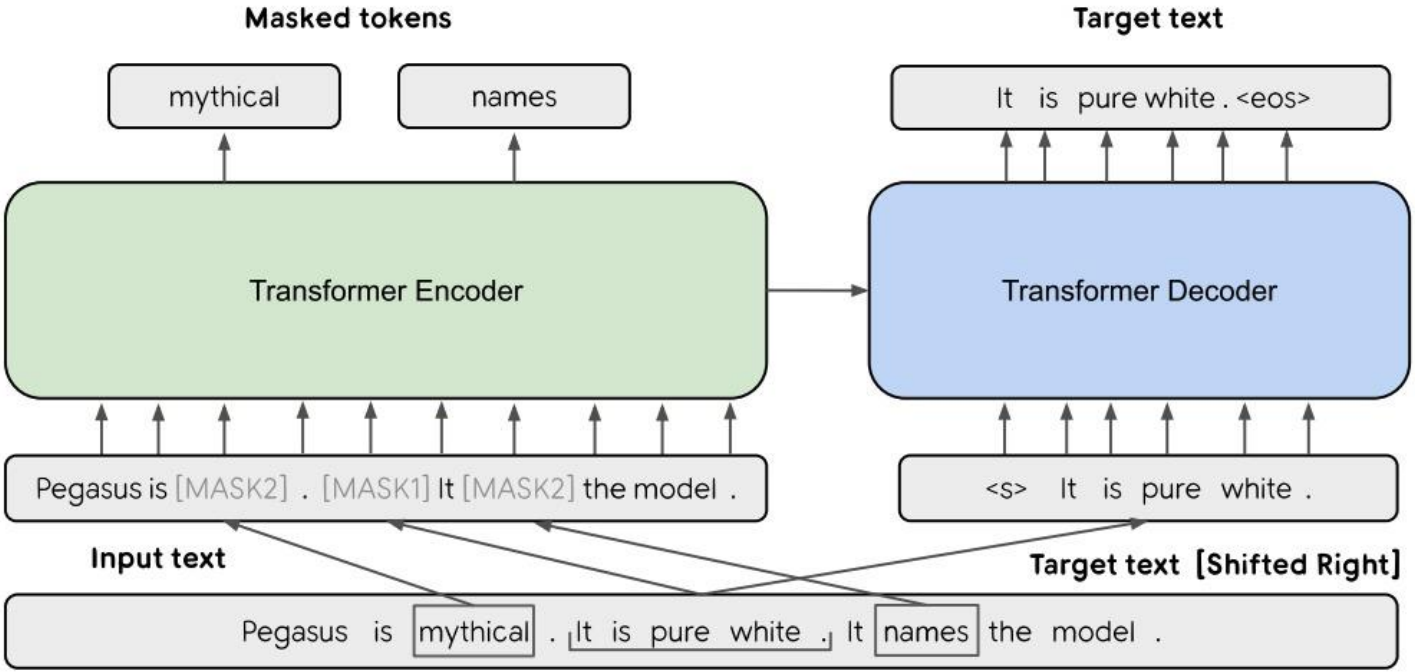

Fig. 2 The base architecture of PEGASUS is a standard Transformer encoder-decoder.

UI Integration

- Flask for Backend
- React.js for Frontend

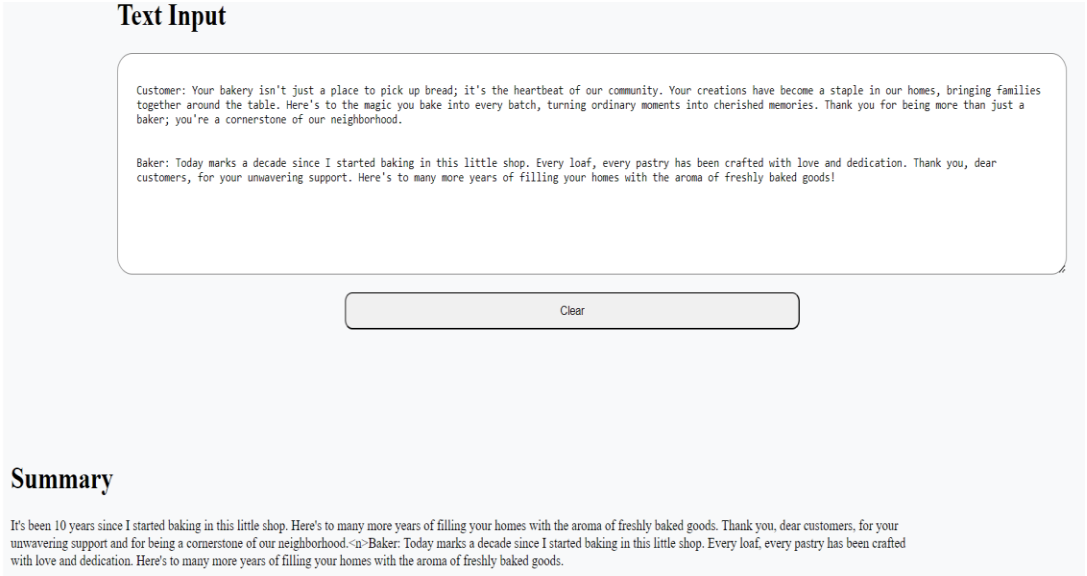

Fig. 3 UI

Evaluation Metrics

	Rouge - 1	Rouge - 2	Rouge - L
Recall	26.8%	18.2%	24.8%
Precision	19.3%	10.4%	17.3%
F1 score	19.3%	12.7%	22.3%

Table 1 - ROUGE metrics evaluation for text

Results & Analysis

- Model Performance. The PEGASUS model was evaluated using the SAMSum dataset, showing effective performance across different text lengths, indicating robustness and adaptability.
- ROUGE Scores. The evaluation was primarily based on ROUGE scores, which assess the overlap between the machine-generated summaries and human-made reference summaries.
- Quality of Summaries. The generated summaries significantly reduced the word count while maintaining the essential meanings of the original texts. This indicates that the model successfully captures the core information while generating concise content.

Limitations

- Distribution of document lengths and complexities in the Samsun dataset may not reflect real-world variability.
- SAMsun Dataset doesn't cover a wide range of domains.
- Fine-tuning Challenges.

Fig. 4 Summarization of Random Text

Conclusion

- Leveraging transformer models like Pegasus significantly enhances abstractive summarization accuracy. Through pretraining on extensive corpora and fine-tuning on specific tasks, Pegasus generates high-quality summaries.
- The SAMSum dataset emerges as a pivotal resource for evaluating and training summarization models. Annotated with human-written summaries, it offers invaluable insights into model performance.
- Generating summaries of the conversational data without many factual errors.
- Despite advancements, challenges persist in handling lengthy documents and capturing nuanced context accurately. Fine-tuning and hyperparameter adjustments are crucial to address these challenges effectively.

Future Work

- Multimodal Summarization
- Domain-specific Summarization
- Interactive Summarization Systems
- Cross-lingual Summarization
- Ethical and Bias Considerations
- Integration with chatbot system

References

1. Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. arXiv:1912.08777 [cs.CL]
2. Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. arXiv:1602.06023 [cs.CL]
3. Joel Neto, Alex Freitas, and Celso Kaestner. 2002. Automatic Text Summarization Using a Machine Learning Approach, Vol. 2507. 205–215. https://doi.org/10.1007/3-540-36127-8_20
4. Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. arXiv:1905.03197 [cs.CL]
5. Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461 [cs.CL]
6. Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. arXiv:1602.06023 [cs.CL]
7. Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. arXiv:1905.02450 [cs.CL]

Thank you!

Questions?