AIM 829 – Natural Language Processing Final Project –

Auto-Generated Research Paper Abstracts

Abheet Sethi MT2024004

Bhatt Purnendu Deven MT2024031

Chirag Sameer Date MT2024034

Yadav Roshan Rajesh MT2024169

22 April 2025

1. **Title**

Auto-Generated Research Paper Abstracts

1. **Problem Statement**

The project aims to automate the generation of research paper abstracts using advanced AI and NLP techniques. Given the increasing length and complexity of research articles, manual summarization is inefficient. Automatically generated abstracts help researchers quickly understand the core insights of a paper, improving both accessibility and searchability. The core challenge is to generate concise, coherent, and citation-aware summaries that maintain the structural integrity and context of the original document.

1. **Objective**

* Extract and clean scientific papers from ArXiv.
* Preprocess content using ScispaCy and custom heuristics.
* Train an encoder-decoder architecture with attention.
* Evaluate summaries using ROUGE and perplexity.
* Automate inference and report generation.

1. **System Architecture**

**Components:**

**Data Source:** ArXiv HTML pages

**Preprocessing:** ScispaCy, regex, custom token filters

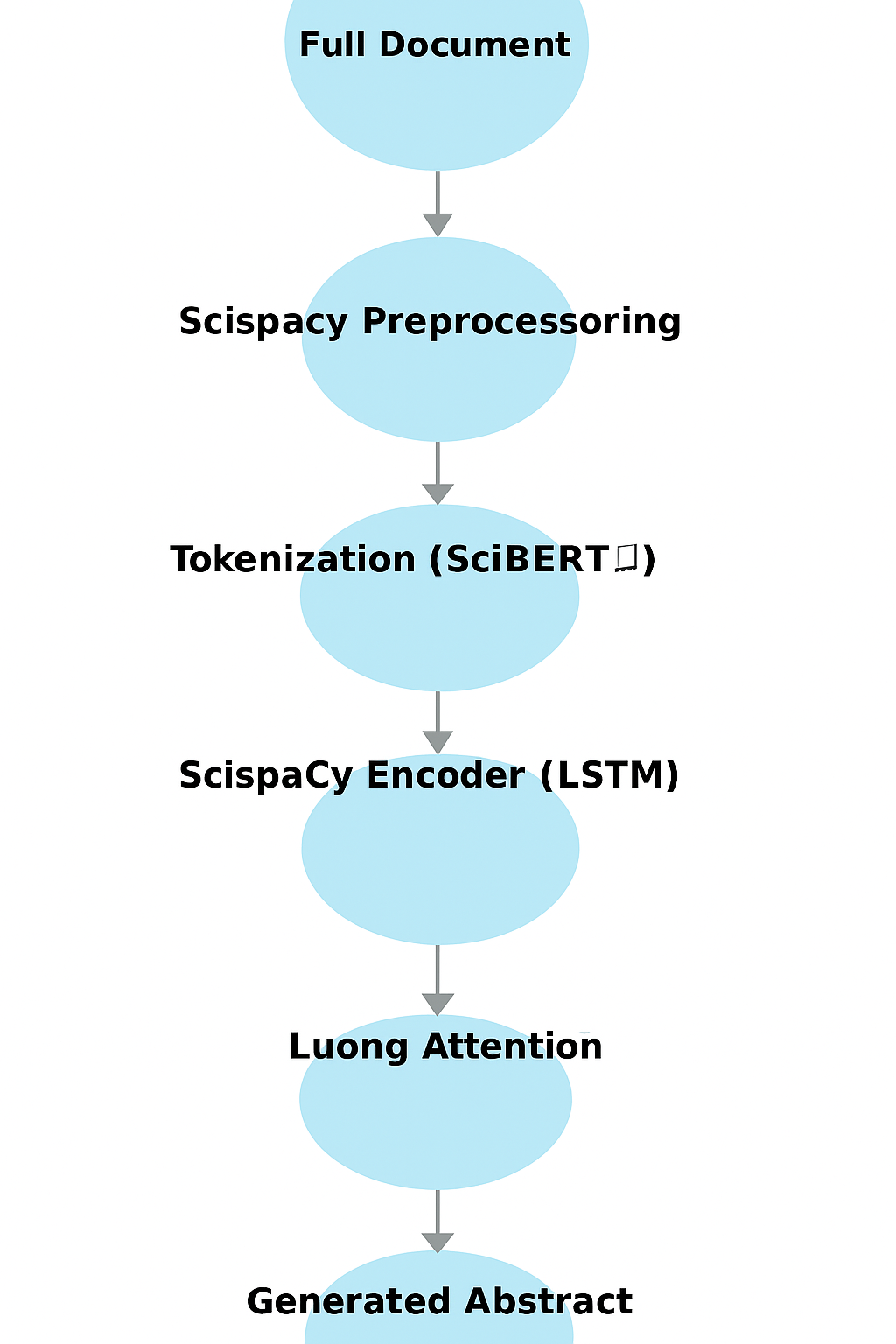
**Encoder:** ScispaCy-based LSTM

**Decoder:** LSTM with Luong Attention

**Training Framework:** PyTorch

**Tokenizer:** SciBERT tokenizer

**Evaluation:** ROUGE, perplexity



1. **Experiments conducted**
   1. Tried using manually trained custom\_tokenizer on the research papers data itself and used pretrained scibert tokenizer.
   2. Training in batches of papers.
   3. Data extraction from downloaded pdfs and html downloaded content and compared data quality using visual scrutiny.
2. **Data Pipeline**

**Paper Extraction**

**papers = extract\_hrefs\_from\_url\_by\_title()**

**download\_arxiv\_html(link, 'paper.html')**

* Uses ArXiv's listing pages to gather links to recent publications in selected domains (e.g., cs.AI, q-bio, etc.).
* The function extract\_hrefs\_from\_url\_by\_title() targets papers that offer an HTML version.
* Each paper's HTML is downloaded using download\_arxiv\_html(), cleaned, and saved for further processing.

**Abstract and Body Extraction**

**abstract, html\_without\_abstract = extract\_abstract(html\_data)**

**main\_paper = extract\_paragraph\_text(html\_without\_abstract)**

* **extract\_abstract()** locates the **<div class='ltx\_abstract'>** tag in the HTML to isolate the abstract.
* Once extracted, the abstract is removed from the HTML to prevent it from being used during training.
* **extract\_paragraph\_text()** then pulls all **<p class='ltx\_p'>** paragraph elements which contain the main body text.
* This separation ensures the training data and the target abstract are distinct, avoiding data leakage.

**Text Cleaning with ScispaCy**

**cleaned\_paper = preprocess\_with\_scispacy(main\_paper)**

* Applies advanced token filtering and lemmatization using ScispaCy.
* Removes:
  + Stopwords and punctuation
  + Irrelevant tokens like URLs, math symbols, or bracketed items
  + Non-alphanumeric words
* The final cleaned text is lowercased and lemmatized, providing high-quality input for the encoder.
* This step enhances model focus and reduces noise, especially valuable in scientific domains with complex vocabulary.

1. **Model Architecture**

**Tokenizer: SciBERT**

**custom\_tokenizer = AutoTokenizer.from\_pretrained("allenai/scibert\_scivocab\_uncased")**

* A domain-specific tokenizer designed for scientific text.
* Based on the SciBERT model trained on a large corpus of scientific publications.
* Supports special tokens like [CLS] (used as BOS) and [SEP] (used as EOS).

**Encoder: ScispacyEncoder**

**class ScispacyEncoder(nn.Module)**

* Accepts raw scientific text.
* Uses ScispaCy to compute token-level embeddings.
* Scores and ranks sentences using a frequency-based scoring system.
* Extracts top-K sentences and converts them into fixed-size embeddings.
* Embeddings are passed through an LSTM to generate encoder outputs and hidden states.

**Decoder: ScispacyDecoder**

**class ScispacyDecoder(nn.Module)**

* Takes input IDs for the target abstract.
* Uses a trainable embedding layer and an LSTM decoder.
* Employs **Luong Attention** to align the current decoding state with encoder outputs.
* Combines context vectors and decoder outputs to predict the next token using a softmax layer.
* Can perform step-by-step generation during inference.

**Seq2Seq Wrapper**

**class SciSummarizationModel(nn.Module)**

* Combines the encoder and decoder into a single model class.
* **forward()** method is used during training with both source and target inputs.
* **generate()** method is used during inference to produce the abstract token by token.

1. **Results**

**Averaged Over 3 Papers**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| **Test Loss** | 4.0502 |
| **Perplexity** | 351.6523 |
| **ROUGE-1** | ~0.2361 |
| **ROUGE-2** | ~0.0243 |
| **ROUGE-L** | ~0.1245 |

**Sample Comparison (Paper 1)**

**📝 Generated Abstract:**

**we present a novel approach to integrating scientific knowledge into generative models, enhancing their realism and consistency in 3d point clouds. the key challenge is that the maximum space is used for enhancing large vision - language models ( lvms ) and neural network - based inference algorithms, which produce the effectiveness of transformer - based models, which suffer from low - quality data curation, which limits their applicability. to address this limitation, we propose a novel unsupervised anomaly detection framework, named quantized - supervised learning with a forget**

**------------------------------------------------------------**

**📄 Original Abstract:**

**Recent advances in protein backbone generation have achieved promising results under structural, functional, or physical constraints. However, existing methods lack the flexibility for precise topology control, limiting navigation of the backbone space. We presentProtPainter, a diffusion-based approach for generating protein backbones conditioned on 3D curves. ProtPainter follows a two-stage process: curve-based sketching and sketch-guided backbone generation. For the first stage, we proposeCurveEncoder, which predicts secondary structure annotations from a curve to parametrize sketch generation. For the second stage, the sketch guides the generative process in Denoising Diffusion Probabilistic Modeling (DDPM) to generate backbones. During this process, we further introduce a fusion scheduling scheme, Helix-Gating, to control the scaling factors. To evaluate, we propose the first benchmark for topology-conditioned protein generation, introducing Protein Restoration Task and a new metric, self-consistency Topology Fitness (scTF). Experiments demonstrate ProtPainter’s ability to generate topology-fit (scTF>>>0.8) and designable (scTM>>>0.5) backbones, with drawing and dragging tasks showcasing its flexibility and versatility.**

**------------------------------------------------------------**

1. **Conclusion**

This project demonstrates that using domain-specific tokenization (SciBERT), ScispaCy embeddings, and attention-enhanced LSTM models, it’s possible to generate meaningful summaries using large number of scientific papers.

1. **Team Member Contribution**

**Abheet Sethi (MT2024004):** Model training loop, checkpointing, loss, Evaluation and Metrics Analysis.

**Purnendu Bhatt (MT2024031):** Model architecture implementation (Encoder, Decoder)

**Chirag Date (MT2024034):** Web scraping from ArXiv, HTML parsing, and data extraction pipeline.

**Roshan Yadav (MT2024169):** Preprocessing module using ScispaCy, text cleaning functions, tokenizer integration.