

**Scientific Summarizer** **for Augmented Productivity**

Date of Submission: 29-Jul-2020

Submitted By:

|  |
| --- |
| ALVES MARTINS HUGO |
| ANG SHANDI |

**Table of Contents**

[Abstract 3](#_Toc36021695)

[Acknowledgement 4](#_Toc36021696)

[1. Background 6](#_Toc36021697)

[2. Methodology and Design 7](#_Toc36021698)

[3. Findings 8](#_Toc36021699)

[4. Evaluation and Analysis 9](#_Toc36021700)

[5. Results 10](#_Toc36021701)

[Conclusion 11](#_Toc36021702)

[Recommendations 12](#_Toc36021703)

[Appendices 13](#_Toc36021704)

[References 14](#_Toc36021705)

# Abstract

The *Abstract* section covers the high level of the project idea and insights into various components of the solution you are developing.

Scientific articles can be annotated with short sentences, called highlights, providing readers with an at-a-glance overview of the main findings. Highlights are usually manually specified by the authors. This report presents a supervised approach based on using state-of-the-art pretrained transformer-based model by Google, Bidirectional Encoder Representations from Transformers (BERT), to achieve the task of automatically extracting important phrases of Scientific articles. The experiment results based on using benchmark articles from ScisummNet, show that the summarizer application has perform on average range on the competition scale.

# Acknowledgement

The *Acknowledgement* section gives a note of thanks in recognition of the project inputs, governance, or any other support provided by key stakeholders involved in the project.

We would like to thank Timothy Liu and Zhang Sheng from Nvidia Singapore Development Pte Ltd for their technical assistance advice and guidance throughout the project. Our project supervisor/mentor, Poh Keam has also been extremely supportive in providing the necessary resources and scoping for this project.

**Table of Contents**

[Abstract 3](#_Toc36021695)

[Acknowledgement 4](#_Toc36021696)

[1. Background 6](#_Toc36021697)

[2. Methodology and Design 7](#_Toc36021698)

[3. Findings 8](#_Toc36021699)

[4. Evaluation and Analysis 9](#_Toc36021700)

[5. Results 10](#_Toc36021701)

[Conclusion 11](#_Toc36021702)

[Recommendations 12](#_Toc36021703)

[Appendices 13](#_Toc36021704)

[References 14](#_Toc36021705)

# 1. Background

Text summarization is a process in which it is abbreviating the source script into a short version maintaining its information content with its original meaning. It is an impossible or difficult task for human beings to summarize very large number of documents by hand. As a contribution to NVDIA’s Artificial Intelligence Research Assistant “AIRA” research project by creating an **AI based Scientific Summarizer research assistant** to work alongside knowledge users with their intellective tasks. The aim of the Research Assistant is to **augment the academic research process,** which is highly complex and non-linear, by facilitating **focused attention** and **time savings** when the user is searching for relevant content.

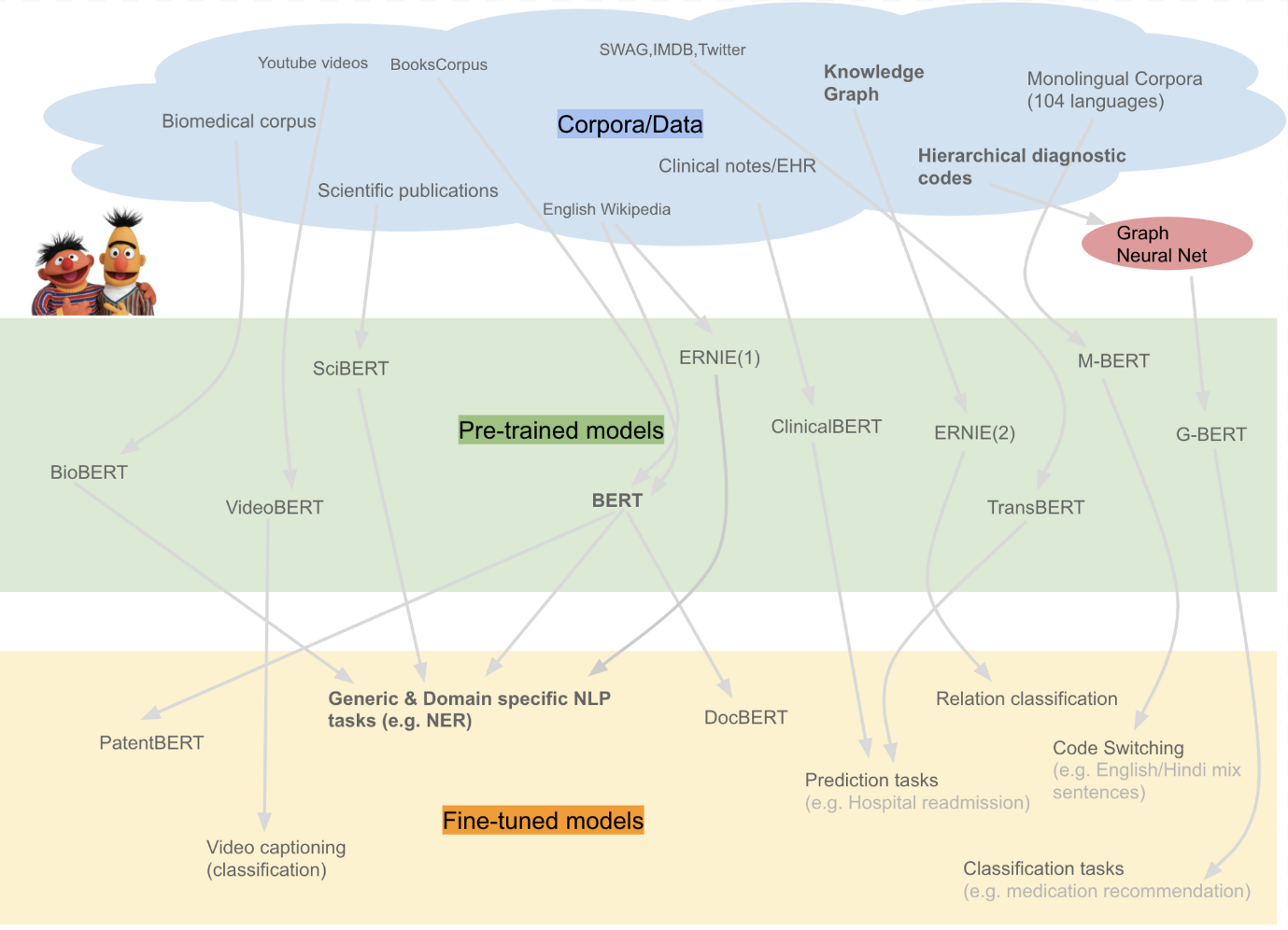
# 2. Methodology and Design

Utilizing the BERT language model, our summarizer computes a contextual representation, i.e. an n-dimensional vector, for every sentence. It applies a hierarchical clustering algorithm to find multiple groups of sentences, such that those sentences nearby in the vector space fall into the same cluster. The summarizer uses the contextualized embeddings to quantify the informative content of sentences and assess the similarity between them. The idea is that those sentences within the same cluster share similar context. Subsequently, the summarizer selects the most informative sentences of each cluster to generate the final summary.

For the application design, we use Streamlit to design the frontend web application for user to input 3 types of text, URL, XML formatted file or TXT files. We also use Streamlit to host the backend server. The summary engine is run by implementing SOTA model, Sci-BERT and Distil-BERT. The inputs will pass through the BERT model and generate a highlighted summary base on the original corpus and the generated summary will be displayed on screen. We also add a voice synthesizer function to read the generating summary for convenience. For evaluation we used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scoring system to compare our application performance with the SciSummNet

# 3. Findings

 Text summarization in NLP can be separated to 2 categories from the point of view of summarization output type, **Extractive text summarization** and **Abstractive text summarization**. Where Extractive summarization choose representative sentences in documents and Abstractive summarization creates summary from nothing. But due to the limitation of hardware computing power generating the abstractive summary, generally extractive approach is used for summarization.



However, during our initial experiment, we explored about BERT’s subsidiaries trained by a famous NLP startup HuggingFace, SciBERT and DistilBERT. SciBERT is a BERT based model but pre-trained in a **large multidomain scientific corpus from the Semantic Scholar dataset** consisting of 1.14 million scientific publications from **computer science** and **biomedical research.** For more General content, we use DistilBERT. DistilBERT is a smaller and faster version of BERT trained by distilling the Bert base. It has 40% less parameters, runs 60% faster while preserving over 95% of Bert’s performances. This distinction between scientific and general type of content is important because the researcher often needs to alternate between scientific articles and other sources of information, such as news, company info, or Wikipedia for instance. Giving the choice to alternate between the two summarizer engines improves the functionality and usability of the app for the researcher

# 4. Evaluation and Analysis

# 5. Results

# Conclusion

The *Conclusion* section of the reporting template gives the main cause(s) of the problem or opportunity highlighted by the project. It provides a summary of the discussions made in the Key Chapters, outlines the main findings, and gives a reference to the Recommendations report section.

With the dramatic growth of the Internet, people are overwhelmed by the tremendous amount of online information and documents be it generic or scientific. This project showcased the extractive summarization of scientific and generic text. Having the capability of summarizing both generic and scientific text, this summarizer engine can solve light-weighted summary problems. Due to the base model being plug and play into our engine, there are limitations of summarizing professional research articles. There is also limitation in Extractive summary as compare to Abstractive summarization. Due to lack of GPU resources and a lot of parameters to tune we end our research on extractive summarization using SciBERT base model. The alternative is to use abstractive implementations and training the model to specialise on scientific summary and tweak some parameters to see whether the results get better.

# Recommendations

The *Recommendations* section of a sample projector report is designed to call people to action based on the findings, analyses and results that have achieved upon project completion. It should be written with actionable and specific sentences and give a solution for the problem investigated by the project.

Some possible future improvements that we recommend is to implement feedback feature to capture the user comments on the quality of summary, this information can then be used to fine tune the model parameters on the final output layers to potentially create a specialize summarizer just for scientific research paper. Another feature can be to create an easy sharing function that allow sharing to other people through different channels like email or messaging apps. The deployment on other platforms like mobile app version with enhanced audio features to allow commuters to listen the summaries. Also, with the release of OpenAI’s new NLP model, GPT3 trained with over 175 billion parameters, future improve to leverage on this new model can significantly increase the productivity of this application.

# Appendices

The *Appendices* section of a typical project reporting document includes any other material in support of project findings. Gantt Charts, Resource Usage Diagrams, Project Schedule Template, Issue Log.

Text summarization: <https://towardsdatascience.com/comparing-text-summarization-techniques-d1e2e465584e>

Some ideas about few-shot learning (potentially useful for thinking about how to finetune the model on user annotations): [https://maelfabien.github.io/machinelearning/NLP\_5/#](https://maelfabien.github.io/machinelearning/NLP_5/)

Pretrained model on scientific articles: <https://github.com/allenai/scibert>

Some interesting ideas in transfer learning, and a potentially useful model: <https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html>

Derek miller research paper on leveraging BERT for extractive text summarization on Lectures :<https://arxiv.org/abs/1906.04165>; git hub repo: <https://github.com/dmmiller612/bert-extractive-summarizer>

Ed Collins, Isabelle Augenstein, Sebastian Riedel. [A Supervised Approach to Extractive Summarisation of Scientific Papers](https://arxiv.org/abs/1706.03946). To appear in Proceedings of CoNLL, July 2017.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, Dragomir R. Radev . 2019. A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks. In ScisummNet. <https://arxiv.org/pdf/1909.01716.pdf>

# References

The final section of your report covers all the sources of information used to make the research and develop the document.