## Supplementary_material_appendix_3.

## SOFTWARE & CODE

The purpose of this section is to provide a clear and complete understanding of the tools and techniques employed, thus facilitating the replicability and validation of our results by other researchers. We utilized various software libraries and tools that brought distinct advantages to our research work, enabling efficient data manipulation, robust analysis, and insightful visualizations. All the resources and software will be made available on a GitHub repository, ensuring the reproducibility of the research. However, we have not yet included the link to ensure anonymity, but it will be included if the paper is accepted. Instead, for this preprint, it is included in the supplementary file (as a zip file), not for review (to ensure anonymity again), but to enable reproducibility should the reviewers ask for it.

Python, a high-level and general-purpose programming language, was the main programming language used in this project. Its wide range of libraries and modules tailored to scientific computing and data analysis greatly facilitated the implementation of our research. An integral part of the Python ecosystem used in this research is Jupyter Notebook, an open-source web application that enables the creation of documents containing live code, equations, visualizations, and narrative text.

One of the key libraries used was **os**, which provided a convenient and reliable way of interacting with the operating system. Given the extensive volume of files that needed to be managed during this research, os was instrumental in enabling seamless navigation, reading, and writing of files, consequently improving the efficiency of data ingestion and preprocessing.

In the field of NLP, the **Natural Language Toolkit (NLTK)**[1] played a critical role. It provided the necessary tools for tokenization, lemmatization, and stop word removal, which are fundamental preprocessing steps in text data analysis.

For Optical Character Recognition (OCR), the **pytesseract**[2] library was utilized. It transformed scanned PDFs and images into machine-readable text, thus making a significant amount of data accessible for subsequent NLP tasks.

A substantial part of our research work was the application of state-of-the-art machine learning models to our dataset. Here, the **Transformers**[3] library by Hugging Face was indispensable. It offered access to a wide range of pre-trained models, including BERT and RoBERTa, which were used extensively in our research. The library's user-friendly API and comprehensive documentation made it possible to quickly fine-tune and adapt these complex models to our specific needs.

The models were deployed for the inference process using **PyTorch**[4]. With its dynamic computational graph and efficient memory usage, PyTorch allowed us to run our models smoothly and interpret the results more intuitively, effectively aiding the inference process.

For visualizing the analysis and results, the **Seaborn**[5] library was used extensively. Seaborn is a Python data visualization library based on Matplotlib, which offers a high-level interface for drawing attractive and informative statistical graphics. With Seaborn, we were able to generate a variety of compelling visualizations that effectively conveyed our research findings.

In summary, the range of libraries and tools employed in this research work significantly streamlined the data manipulation, model building, analysis, and visualization tasks, enabling us to focus more on the core research objectives.

---

[1] For more information on NLTK library: https://www.nltk.org/

[2] For more information on pytesseract library: https://pypi.org/project/pytesseract/

[3] For more information on Transformers library: https://huggingface.co/docs/transformers/index

[4] For more information on PyTorch library: https://pytorch.org/

[5] For more information on Seaborn library: https://seaborn.pydata.org/