

# Brief Summary of the model - Policy Summary Assistant

Sucheti Kanere, Abhishek Sarodaya

Project Report

## Use Case Title

AI Policy Summary Assistant for Insurance Documents

## Category

Artificial Intelligence (AI) / Natural Language Processing (NLP)

## Technology Stack

- **Python:** Core language that handles AI processing, NLP logic, and system coordination.
- **Streamlit:** Framework that drives the interactive web interface for uploads and visualization.
- **Hugging Face Transformers:** Summarization model (BART) pre-trained for policy summarization.
- **PyTorch:** Backend engine that allows models to run efficiently on GPU.
- **Regular Expressions (Regex):** Used to extract entities such as policy number, premium, and dates.
- **TextStat:** Used to calculate readability of the generated summary.
- **AsyncIO:** Handles asynchronous summarization of text chunks for speed improvement.
- **NLP & ML Concepts:** Enable smart summarization and compliance understanding in the insurance domain.

## Brief Summary of Model

Policy Summary Assistant is an artificial intelligence based document intelligence platform that summarizes, analyzes and interprets complex insurance policy documents. The system leverages state-of-the-art Natural Language Processing (NLP) techniques and Transformer-based Large Language Models (LLMs) to produce summaries that are short, contextually relevant, and human-like. The model aims to provide answers to the time-consuming problem of manual policy review, by allowing insightful extraction, compliance checking and metadata recognition in the insurance industry in seconds.

The application design is based on a modular and a multi-tiered system. The Frontend layer is developed by Streamlit's python library with a simple visual output at uploading the document, seeing the summary and finally downloading results. The backend intelligence tier (written in Python) coordinates text extraction and cleaning as well as an asynchronous text processing stage via the asyncio module. The core summarization is Hugging Face Transformers based using pytorch, with the model `sshleifer/distilbart-cnn-12-6` - a distilled version of Facebook's BART. It is still an encoder-decoder transformer model and is capable of performing abstractive summarization which involves generating

new text that is semantically coherent rather than merely extracting key sentences.

Processing pipeline starts with document ingestion in which raw text is extracted from PDF, Word files using python modules PyPDF2 and python-docx. The text is normalized in a regex based way and then smart chunked to keep context is segmented into semantically meaningful pieces. Each chunk is independently summarized (async) by the DistilBART model and the outputs are polished into an academic tone. For interpretability, the final aggregate summary is organized into sections such as Introduction, Key Insights and Conclusion.

In addition to summarisation, the system includes a rule-based risk /compliance engine performs a check against critical policy elements such as Liability, Exclusions, and Claims Procedure to identify possible holes. In addition, entities based on regular expressions (e.g., policy numbers ) and other entities (coverage limits, premium amounts, validity periods) are extracted structured metadata.

The summarized content, compliance information, and metadata are available on-line under several well organized tabs using Streamlit and can be downloaded as professional pdf or text reports with FPDF. The end-to-end system illustrates the usefulness of AI and NLP techniques to automate the insurance document analysis for faster processing, better readability, and more accurate decision making. It is a perfect example of how to combine deep learning, asynchronous programming, and software engineering in a way that the end result is a scalable and domain specific AI solution.

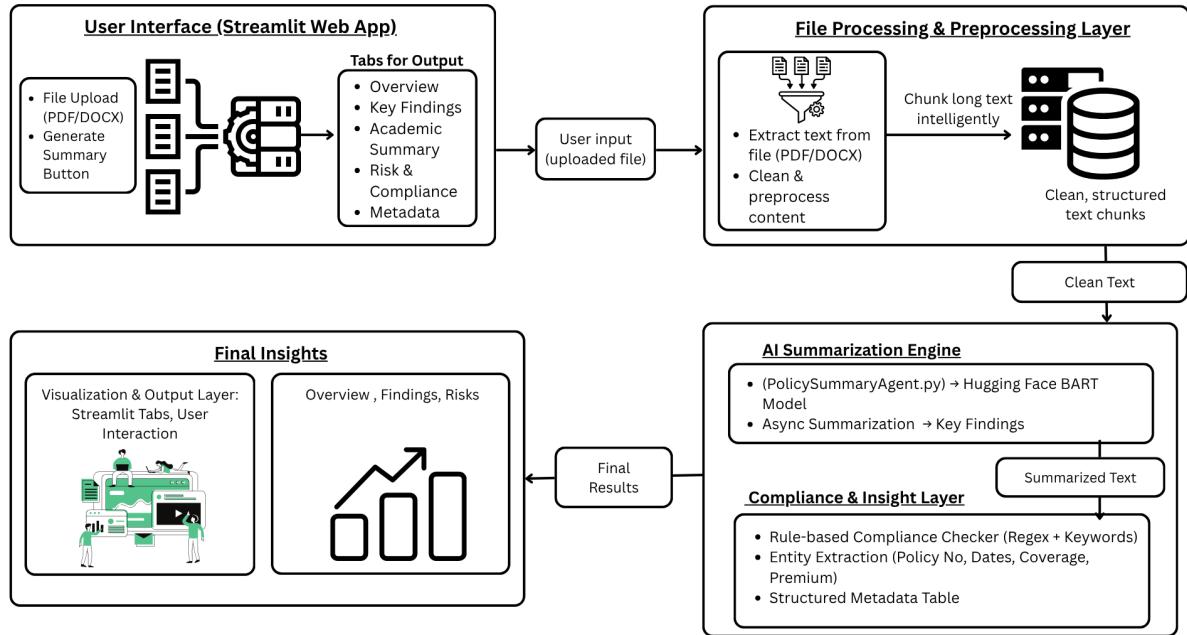


Figure 1: System Architecture

The Figure 1 shows that the system is based on a modular architecture with five layers, from user interface to AI summarization and insight generation. Preprocessed documents are transformed leveraging summarization based on transformers and, then, they are checked for compliance through rule-based procedures generating structured insights. The Streamlit interface displays these results interactively so users can access the summary, key findings, and metadata in real time.