# LEGAL DOCUMENT SUMMARIZATION AND ANALYSIS

TEAM DETAILS:

1. RITU SINHA - ritu.mutual@gmail.com
2. POOJA KUMARI - thisispooja97@gmail.com
3. KRUTHIKA K - kruthikakalmali@gmail.com

GUIDE : SOWMYA B. J

COLLEGE: Ramaiah Institute of Technology, Bangalore

## OBJECTIVE:

To improve the way Legal Documents and Terms and Conditions contracts are presented and analyzed. Provide a comprehensive and easy to understand summary of legal documents and analytics of the text, allowing a deeper insight into it for the reader

## PROJECT DESCRIPTION:

Our provision is two fold. We have primarily worked with Terms of Service documents and Legal Judgements. We thus provide the following:

a) Legal Judgement - Provision of a web application where users can upload a .TXT file of their legal document. We analyze the document and provide:
    - ➔ Comprehensible summary of text
    - ➔ Important Keywords to help understand the text better
    - ➔ Certain informatics such as persons involved, locations, organizations involved etc.
    - ➔ Provision of a bot which can be asked questions to gain better understanding of the document

b) Privacy Document domain - Provision of a plugin which has the ability to find the Terms and Conditions link for a website and then provide an analysis of the complexity and readability of the document. It also extracts the most important sentences from the documents for the user's perusal. Thus this product helps a user be better informed about the conditions they are agreeing to.

## BUSINESS MODEL:

This model can be used extensively in the legal domain. It can be used by legal professionals to obtain a simpler to read document while ensuring that key points are not lost. It also helps in quick perusal and to understand the structure of the case by referring to the people involved, location, etc. This can also be used by an average person to better understand their legal documents. It would specifically help users with respect to understanding a document without the legal jargons involved.

## TECHNICAL SPECIFICATION:

a) Legal Document Summarizer

Dataset Used: The dataset for determining the category of the document was obtained by crawling through the documents available in https://indiankanoon.org/browse/ which consists of judgements and tribunal and court specific legal documents.

Algorithm: The method of summarization used here is extractive in nature. The summarization pipeline consists of the following steps:

i) preprocessing - splitting documents into sentences which are stemmed, lemmatized, case-normalized, and cleared of stop words

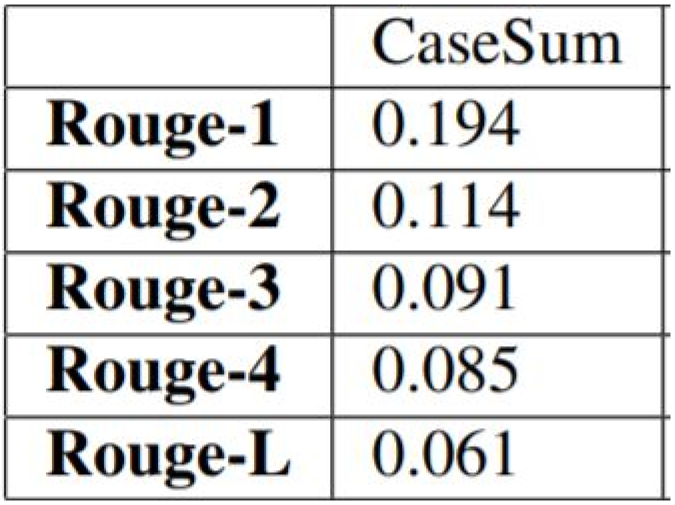
ii) scoring of sentence relevance - Sentences are scored using a TF*IDF matrix built from thousands of legal case reports, which counts term frequency using $TF * IDF_t = TF_t * 1/\log N DF_t$ where N refers to the number of documents, $TF_t$ is the total count of term t, and $DF_t$ is the number of documents in which t appears. These scores are summed over each sentence and normalized by the sentence length. This normalization step ensures the system does not bias long sentences.

ii) domain processing - In order to include additional domain information, we first extract a list of all entities from the text. Parties can be extracted from case titles because of the document structure. Similarly, abbreviations of entity names are identified to aid the reader's understanding of summaries.

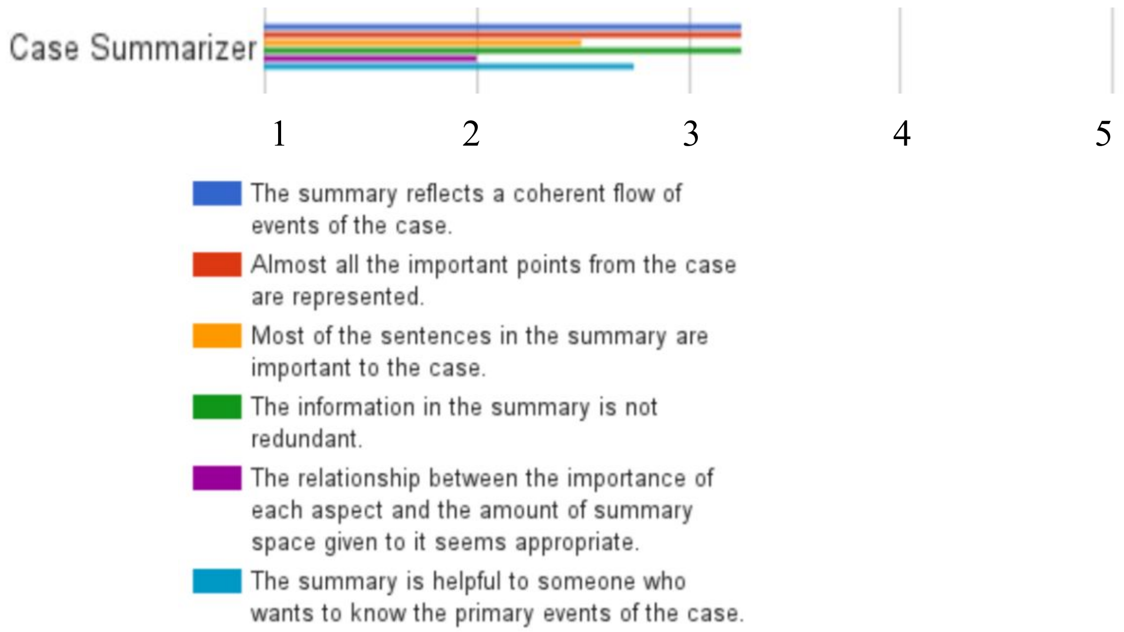
We do not use specific cue words or catchphrases, but adjust sentence scores using occurrences of known entities, dates, and proximity to section headings. The adjustment function is $w(new) = w(old) + \sigma(0.2d + 0.3e + 1.5s)$, where $\sigma$ is the standard deviation among sentence scores, d refers to the number of dates present, e is the number of entity mentions, and s is a boolean indicating the start of a section. (Weights experimentally determined)

We use a multinomial Naive Bayes model for the prediction of category based on the content of the file. NLTK is used for preprocessing and other extraction.

Evaluation: ROUGE package provides a pairwise comparison method for evaluating candidate summaries against human-provided ones. The ROUGE metric has multiple variants and may be applied at the word, phrase, or sentence level. In this case, we used ROUGE-N, which measures the overlap of n-grams between summaries, with N = 1, 2, 3, and 4. We also computed the ROUGE-L score, a metric similar to an F-measure based on sentence-level similarity of two summaries.



| | CaseSum |
|---|---|
| Rouge-1 | 0.194 |
| Rouge-2 | 0.114 |
| Rouge-3 | 0.091 |
| Rouge-4 | 0.085 |
| Rouge-L | 0.061 |



<u>User Interface</u> : We use the Django framework to create a web application. Here users can upload their file and it will be studied to give the required summary and informatics.

The file is directly processed and not stored on any server. For the bot, we use the Microsoft LUIS framework, and we have currently trained the bot on minimum information. Thus the user can query the bot for their requirements.

<u>Tech Stack</u> :

| | |
|---|---|
| HTML JAVASCRIPT CSS Bootstrap | This is used for the frontend for the web page and to provide the required functionality |
| LUIS | Language Understanding (**LUIS**) is a cloud-based API service that applies custom machine-learning intelligence to a user's conversational, natural language text to predict overall meaning, and pull out relevant, detailed information.This is used for the chatbot framework and is trained on specific arguments |
| | Microsoft Bing Speech API is a cloud-based API that provides advanced algorithms to process spoken language, it allow developers add speech driven actions to their applications including real-time interaction with the user.This is used for Speech to Text and Text to Speech, thus enabling the bot to work with voice as well |
| django | It is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern. We use this for development of our application. |

| | |
|---|---|
| **Microsoft Program Synthesis using Examples SDK** | Given a domain-specific language (DSL) and input-output examples for the desired program's behavior, PROSE synthesizes a ranked set of DSL programs that are consistent with the examples.This is used for the key highlights of the document. |
| **NLTK, SKLEARN, WORDNETLEMMATIZER, MULTINOMIALNB** | Libraries used for creating the model |

Final product:

Further work:

- ➔ We would like to extend the file formats supported to pdf etc
- ➔ We would like to train the bot for specific user examples so it is able to answer any document query
- ➔ We would like to explore the idea of using a network for abstractive legal summarization
- ➔ We would like to expand our training set so our model can handle different types of data

b) Privacy Document Analyzer:

Dataset used: We have used the terms and conditions documents scraped from the web for creating annotated sets with the score for each phrase.

Algorithm:

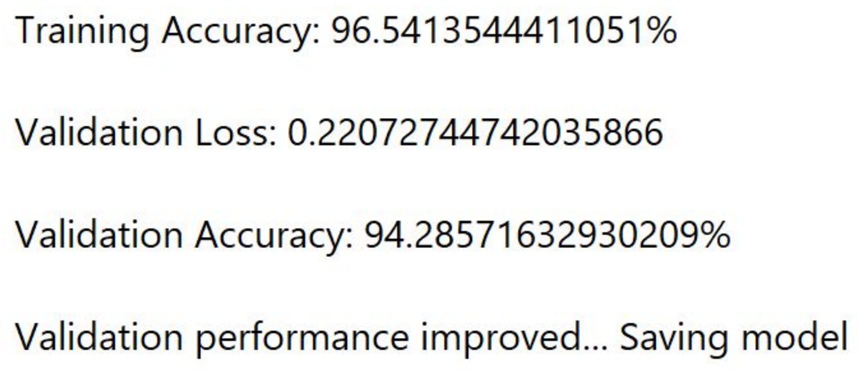
a) Extractive Summarization of the T&Cs and policies is done using the unsupervised TextRank summarization algorithm to identify the key sentences in the document.

b) We also carry out Text Classification of the document's sentences, similar to Sentiment Analysis, to recognize the polarity of a given sentence in the policy. This was done using a Transformer model (BERT) with a logistic regression head fine-tuned for our domain-specific task using a labelled dataset.

c) Readability and complexity scores of the document are calculated and assigned using a publicly available API which uses the Flesch–Kincaid Grading method.

d) Finally, we also used a pre-trained Named Entity Recognition model to identify key entities in the document that should be brought to the consumer's attention such as organisations involved and actions that the policy entails.

Evaluation: After 8 epochs, we obtain the following result:

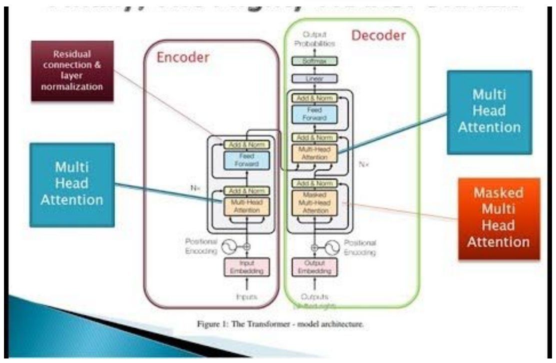


```
Training Accuracy: 96.5413544411051%

Validation Loss: 0.22072744742035866

Validation Accuracy: 94.28571632930209%

Validation performance improved... Saving model
```

User Interface: The backend is a Flask server which communicates to our front end using REST API. On the client side, a userscript using jQuery was created for use in a manager such as Tampermonkey . A button is created in the top right corner of the page which will, when clicked, filter through all the <a> tags on the page to find the link to the privacy policy page. When the page is opened, the script will confirm that the user is on the right page and then send an http GET request to the backend with the page URL. The response from the server will be parsed and displayed to show the summary along with some ratings for the policy through responsive design.

Tech stack:

| | |
|---|---|
| | This was used for the frontend for our application |
| | TamperMonkey was used as the userscript to create the required plugin for our application |
| | Flask is used for the backend server which communicates with the front-end through REST API |

| | |
|---|---|
| PYTORCH | Pytorch was used for the creation of the Transformer Model and extractive summarizer |
|  Residual connection & layer normalization<br>Encoder<br>Decoder<br>Multi Head Attention<br>Multi Head Attention<br>Masked Multi Head Attention<br>Output Probabilities<br>Softmax<br>Linear<br>Add & Norm<br>Feed Forward<br>Multi-Head Attention<br>Masked Multi-Head Attention<br>N×<br>Positional Encoding<br>Input Embedding<br>Output Embedding<br>Inputs<br>Outputs (shifted right) Figure 1: The Transformer - model architecture. | BERT is used as a transformer model, fine tuned to the specific domain and thus it helps us better understand legal data |
| NLTK<br>SPACY<br>TRANSFORMERS | Python libraries used for our model |

<u>Further work:</u>

➔ generate a more detailed dashboard which may help users to better visualize what happens to their data

➔ Abstractive Text Summarisation, which can accurately paraphrase complex legalese into easy-to-read sentences

➔ Making use of more advanced NLP techniques could refine our summaries, using commonly-used lexicon and even colloquialisms to make our summaries very user-friendly