ML Presentation - Project Sankshepika

Team
Swetha M. (23M0756)
Rashmi Kokare (23M0785)
Nikita (23M0807)
Akanksha Dadhich (23M0830)

Task Description

Legal texts often contain **complex language, specialized terms, and long sections,** which can overwhelm people without legal knowledge.

These documents are condensed into clear, straightforward summaries by summarizers. Key points are extracted, **allowing non-experts** to understand vital details without going into the complexities.

Moreover, time is saved, and the need for costly legal consultations is reduced by legal summarizers.

This promotes **access to legal information** and encourages a fair and just legal system.

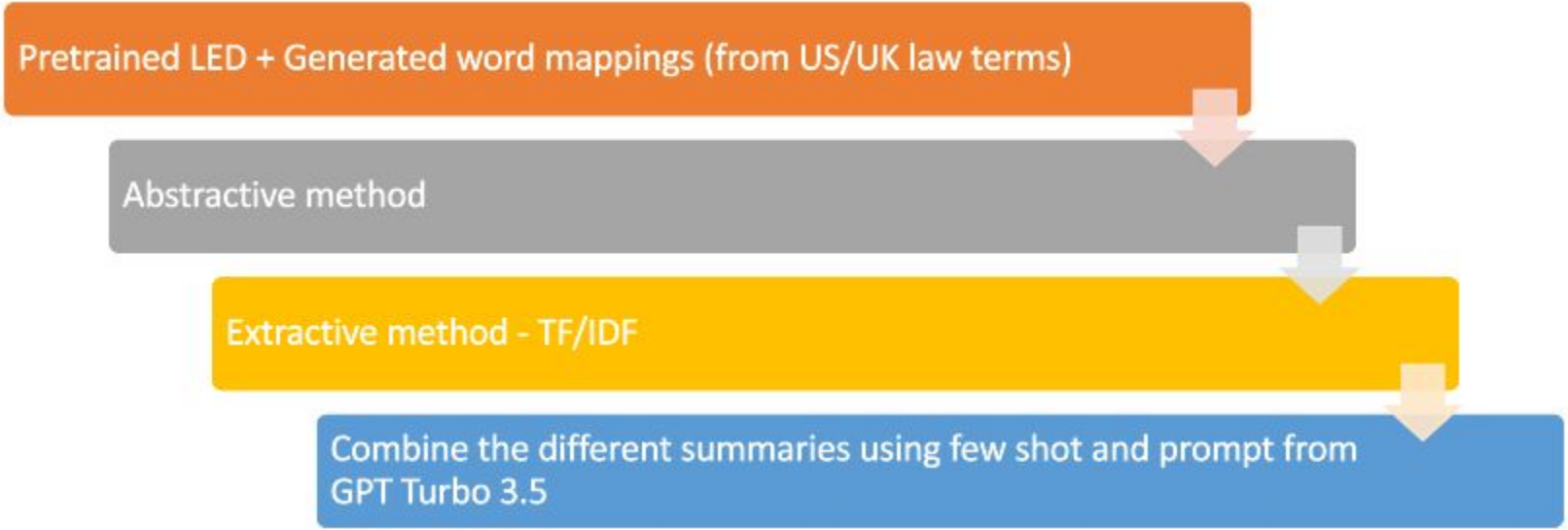
Challenges Addressed

1. The pretrained model vocabulary is **predominantly US based**. “US attorney office” pops up in Indian dataset summaries
2. The LLMs which can take in prompts **cannot handle token limits** ie. the size of the legal documents
3. The extractive and other prior methods are **less human understandable.** (preferred by experts)

US Attorney appearing in Indian dataset

prisoner, do not attract the attention of the law. Indeed, victim reparation is still the vanishing point of our criminal law ! This is a deficiency in the system which must be rectified by the Legislature. We can only draw attention to this matter through the use of the internet and the social media. The court also granted the family of the victim of the driver the right to a hearing. In a parallel action, the **U.S. Attorney's Office for the Southern District of New York today announced criminal charges against the driver. As alleged in the complaint, Indian Transport is acquiring a menacing reputation.** More people die of road accidents than by most diseases, so much so the Indian highways are among the top killers of the country. The rising number of road casualties and rising numbers of road fatalities make it particularly important that every State take note of the human price of highway neglect, of State transport

Outline of method



Challenge:

- The fine-tuning of pretrained model (using tokenizer of Pretrained) was very huge to train. It led to repeated crashing of machines where training was done.

Dataset

Link: https://zenodo.org/records/7152317#.Yz6mJ9JByC0

UK-Abstractive dataset (UK-Abs): https://www.supremecourt.uk/decided-cases/

Indian-Extractive dataset (IN-Ext): Indian summaries by 2 law experts A1, A2:

Indian-Abstractive dataset (IN-Abs): http://www.liiofindia.org/in/cases/cen/INSC/

Extractive method vs Abstractive method

Extractive method using TF-IDF

- Finds the most important sentences from the very large document.
- Most useful.

Advantage:

- No new information. Exact same from source

Disadvantage:

- Simplification is not done. It is complex in language and not continuous.

Abstractive Method

- Take pretrained model from Legal-LED
- Generate a mapping for most differing words in the vocabulary of the pretrained vs the current dataset.
- Replace from the vocabulary mapping

Advantage:

- The language is simple, summary is meaningful.

Disadvantage:

- Validity of data is not ensured. Can’t customize according to requirements

Training and Combining all the 3 methods

Transformer based model for summarization

- Use custom dataset
  - INDIAN dataset
- Train the transformer on GPU
  - <GO>, <STOP> <EOS> tokenizations
  - 20 epochs
  - Positional encoding, Multi Headed attention
  - Vocab size - Encoder - 1.3L, Decoder - 67k
- ***# hyper-params***
- ***num_layers = 4***
- ***d_model = 128***
- ***dff = 512***
- ***num_heads = 8***

LLM using prompt engineering and few shot

- Establish context of system and inject the summaries from the previous methods through prompt
- Large documents couldn't fit the token limit. But the individual summaries can.
- **Advantage**: Customize requirements
  - Simplicity, highlight timeline, reducing US related terms etc. in prompt
- Make a call to GPT 3.5 turbo to generate a combination of the 3 methods

Experiment Details and Main Results

SNo	Method/Model	Readability	Usefulness
1	Pretrained LED	Good	Average
2	Pretrained LED + Word mappings	Good	Average
3	Transformer model	Average	Average
4	Extractive model	Average	Good
5	Combined approach of the methods 2, 3, 4	Good	Average

Related Work

The common approaches to solve Summarisation are using Extractive (choose most important lines) and Abstractive (summarize the content in simpler smaller sentences).

Abstractive models:

1. BART
2. Legal-Pegasus
3. Legal-LED .

BART and Pegasus have token limit issues which are solved by a chunking mechanism. LED is a longformer model which can handle large lengthy documents.

Extractive methods: Most extractive methods are unsupervised. They can serve as a good starting point for identifying key important sentences of our test document. Some of the methods available as existing code are:

1. LetSum
2. Gist
3. CaseSummarizer
4. SummaRuNNer (domain independent supervised method)

Conclusion

1. The pretrained models suffer from incorrect or hallucinated information especially from US Law
2. Pre-trained transformer finetuning takes a lot of time and resources
3. Different approaches - Abstractive & Extractive have different pros and cons
4. Combining summaries is easier for LLMs with token limit to get a reasonable output leveraging all methods

Work split up amongst the team members

1. **Pretrained model and its word mappings - Nikita & Rashmi**
2. **Extractive method using TF-IDF techniques - Nikita & Rashmi**
3. Attempts on fine tuning pretrained model (Crashed due to high RAM requirements) - All
4. Study of pretrained model outputs, prior work - All
5. **Trained Abstractive model from a transformer - Akanksha & Swetha**
6. **Merging the summaries using prompt from GPT 3.5 - Akanksha & Swetha**

References to existing code or libraries used for the project.

The codebase from https://github.com/Law-AI/summarization gives extensive base code to explore various Abstractive and Extractive solutions for summarisation.

The codebase for a basic transformer model https://github.com/rojagtap/abstractive_summarizer. This was used for training our model.