Detecting contentious lines in Terms and Conditions using NLP Sentiment Analysis

Introduction

These days, when we access any digital services, we need to agree to lengthy policy documents, commonly known as terms and conditions (T&C). These are legally binding agreements that define the rights, responsibilities and obligations of all parties involved. Since these documents are so long and intricate, they might contain deceptive or unfair clauses that users might miss. Most users tend to skim through or even blindly agree to these lengthy contracts without fully understanding the implications of their consent.

As a result, individuals may end up signing away critical rights or expose themselves to unfavourable conditions. Therefore, there is an urgent need to identify and highlight potentially malicious or deceptive terms so that users are not left unaware and vulnerable to exploitation. Identifying such contentious clauses is challenging due to the complex and ambiguous nature of legal language. Manual reviews are not only time consuming but also prone to errors.

In recent years, Natural Language Processing (NLP) has proven to be a valuable tool for effectively analysing text-based data. Although sentiment analysis, an application of natural language processing technologies, has been widely applied in fields such as customer reviews and social media monitoring, it's application in the legal field remains relatively unexplored.

This paper explores the application of sentiment analysis to detect contentious lines in T&C documents. The goal is to help users become more aware of what they are agreeing to, fostering greater awareness and protection for users.

Literature Review

ToS;DR or Terms of Service; Didn't Read is a combined effort to make a website which analyzes and ranks the Terms and Conditions of many famous services, for example Facebook, Amazon, Spotify, etc. This is the only actual analyzer for these services, with other sites just being simple text summarizers. ToS;DR only shows very limited well-known sites with a tailored database, and it does not offer any service to paste text to summarize. Various other services are present for summarizing text, but they may not give appropriate results with regards to terms and conditions.

AI Summarizer, quillbot, scribbl text summarizer are some examples of such services. Since a lot of legal language is involved, some specific phrases may be misinterpreted or oversimplified. Domain specific services always provide a better result and analysis as to general service. These general interpreters may struggle to understand the context across the multiple sections of a legal document. Such context would be privacy of user data, user rights, obligations, ambiguous statements etc.

Thus, having a model focusing on parsing through and analyzing a terms and conditions document being trained on legal jargon with specific datasets is necessary.

Methodology

Dataset selection

There was a lack of data specific to the purpose of our study. Mainly, we needed something that provides us with very specify sentiment analysis data; the one needed to properly recognize legal terms. The Terms of Service, (aka T&C) document contains terms and sentences that are otherwise not available in normal sentiment analysis datasets, which focus more on normal day-to-day sentences one would hear in a conversation. Due to this, we went ahead to generate our own dataset by manually skimming through the ToS documents of a few companies (disclosed later) and flagging each sentence as "important" or not.

Preprocessing

Manually generated data didn't need much since we ensured that there were no missing values or duplicate records while generating the dataset. The sentences are tokenized using the Tokenizer() class from tensorflow.keras.preprocessing.text module. The dataset has been split into 80%-20% training-testing sentences respectively.

Model & parameter tuning

We experimented with quite a few combinations of different models, namely "Global Average Pooling 1D", "LSTM (Long Short-Term Memory)", and Bidirectional LSTMs. The commonly used "adam" optimizer is used here.

Evaluation Metrics

All models were evaluated on loss, accuracy, and F1 score.

Results and Analysis

Out of these three, the model with a single LSTM layer performed quite bad, with the accuracy and F1 score staying consistent across different epochs, but with a very high loss.

Epochs	Score		
	Loss	Accuracy	F1
10	0.3875	0.8719	0
25	0.3833	0.8719	0.2271
50	0.3829	0.8719	0.2271
75	0.3579	0.8719	2.2272

The Bidirectional LSTM performed better in terms of accuracy and F1 score, but it had an even higher loss, peaking at 0.8413 at 75 epochs.

Epochs	Score		
	Loss	Accuracy	F1
10	0.3762	0.9089	0.6032
25	0.6960	0.8978	0.5668
50	0.6800	0.9002	0.5744
75	0.8413	0.9027	0.5820

On the other hand, the model with the Global Average Pooling 1D layer performed overall the best at 50 epochs, by only having a loss of 0.2067, an accuracy of 0.9310 and F1 score of 0.7403.

Epochs	Score		
	Loss	Accuracy	F1
10	0.3265	0.8786	0.3956
25	0.1903	0.9347	0.7200
50	0.2067	0.9310	0.7403
75	0.2857	0.9298	0.6629

All three models were trained on varying epochs, where all of them usually show promising results at 50 epochs, suggesting that going any higher results in overfitting of the data. This is especially prevalent in analysis of the Bidirectional LSTM model, where the loss jumps up from 0.68 at 50 epochs to 0.8413 at 75 epochs. Naturally, a loss of 0.68 is also significant, but this jump shows how severe overfitting can get.

This also attributes to our inadequate amount of data. Since we had to manually generate the dataset, it's not that large. This results in more complex layers like LSTM and Bidirectional LSTM overfitting on the data too much, which makes them perform poorly, despite them being normally more suitable for applications like these. Due to the unique case of being such a small dataset, a simpler model like GlobalAveragePooling1D performs better here since it does not delve too deep in the patterns of the training sentences, which grants it superior performance over the others.

Conclusion

The application of an NLP model to detect malicious intent in terms and conditions documents represents a significant step toward empowering users and enhancing transparency in legal agreements. By automatically analyzing contract language, the model identifies contentious clauses that compromise user privacy, unfair limitations on liability, or hidden data-sharing practices. This model can drastically reduce the time consumption and effort required by legal practitioners to find ambiguity, ill intent or legal traps within a document.

The goal of such a tool or service is not only to enhance user protection but also to develop a more ethical standard for drafting legal documents.

References

https://tosdr.org

https://www.summarizer.org

https://quillbot.com/summarize

https://ijccts.org/index.php/pub/article/view/176

https://www.sciencedirect.com/science/article/abs/pii/S0957417420305030