Critical Information Extraction from Terms of Services Document

Aditya Ashok Dave
Univ. of Southern California
daveadit@usc.edu

Akanksha Sanjay Nogaja
Univ. of Southern California
nogaja@usc.edu

Lavina Lavakumar Agarwal
Univ. of Southern California
llagarwa@usc.edu

Shreya Venkatesh Prabhu
Univ. of Southern California
prabhus@usc.edu

Sai Sree Yoshitha Akunuri
Univ. of Southern California
akunuri@usc.edu

Abstract

Terms of Services (ToS) are legal agreements between users and service providers. In order for a user to consume any service they must accept the terms. However, since ToS documents are very verbose and use a very opaque jargon, users tend to acknowledge them without fully understanding the agreement. This can lead to the user signing obligations which they might not be willing to in reality, or they might be exposed to unfair terms and practices. The proposed idea is to make users more informed about the unfairness of the clauses in ToS and also present the obligations imposed by it.

1 Introduction

"I have read and agree to the terms and conditions." Many online services require accepting their Terms of Service in order to use them. According to a study (McDonald AM and Cranor LF, 2008), the annual time estimate for a frequent online service user to skim through the terms was found to be 80 hours and 200 hours to read through the entire document. Moreover, users tend to sign the ToS without knowing the intricacies and implications of the same. Hence, a NLP based solution is proposed which provides users with a prompt way of going through the ToS documents by providing them with critical information i.e. automatically detected **unfair clauses** and **obligations** that they must comply with. Thus enabling them to make an informed decision on whether to accept the terms in the ToS document.

The contributions of this project to the earlier research in this area are:

- an extensive comparison of Transformer based embeddings (RoBERTa and XLNet) with various deep learning models for fairness classification.
- considering and identifying user obligated clauses as critical information in addition to unfair clauses.

2 Related Work

CLAUDETTE (Marco Lippi et al., 2018) is an experimental study to detect potentially unfair clauses from ToS of online platforms using machine learning. 7 models (SVM, CNN, LSTM and HMM with various combinations of features) were tested on ToS dataset and the accuracy metrics (Precision, Recall and F1 score) were reported. While most of the models were sentence level, SVM with HMM was used to build a collective classification algorithm to include sequence of sentences. Top 5 models among these were picked to create an ensemble which reported the best F1 score of 0.806. Another research (Guarino A. et al., 2021) found promising results using Universal Sentence Encoder with SVM for classification with an F1 score of 0.87.

In (L. Zheng et al., 2021), the authors have spoken about whether pre-training the neural network model is useful for law and legal dataset processing. For this, they compared BERT base model performance on ToS dataset with BERT Double, a BERT model which was trained for extra 1M iterations and Custom Legal-BERT, a BERT model trained from scratch on Harvard law case corpus. They found that BERT Double, with 77.3% F1 score and Custom Legal-BERT, with 78.7% F1 score, outperformed BERT base model and also the highest performing model from Claudette for the general setting of Terms of Service by 0.4% and 1.8% respectively.

(Vikas and Roshann, 2020), which forms the basis for finding obligatory terms, uses transfer learning to create a secondary dataset using

Named Entity Recognition in English legal documents. For extracting entities, the authors have used libraries such as FlairNLP, AllenNLP, BERT, deeppavlov on the Entity-Recognition-Datasets. It was found that FlairNLP, being trained on multilingual text, outperforms AllenNLP followed by BERT.

3 Datasets

3.1 ToS Data

ToS dataset was created as a part of CLAUDETTE (Marco Lippi et al., 2018) experimental study. The ToS clauses in the dataset are annotated for 8 categories of unfairness using XML mark-up. The categories are documented in Table 1. Each of these categories are further subcategorized into 3 levels: clearly fair, potentially unfair and clearly unfair, giving us 24 classes. Of these 24, annotated data is available for 18 categories and the authors have made it publicly available at `http://claudette.eui.eu/ToS.zip`.

Category	Annotation
Arbitration	<a>
Unilateral Change	<ch>
Content Removal	<cr>
Jurisdiction	<j>
Choice of Law	<law>
Limitation of Liability	<ltd>
Unilateral Termination	<ter>
Contract by Using	<use>

Table 1: ToS clauses categories

3.2 Data Preprocessing

Potentially unfair and clearly unfair clauses were merged into one class - unfair. For data preprocessing, the ToS dataset was subjected to techniques such as converting dataset to lower case and removing accents, html tags, urls, and non alphabetical characters. For Word2Vec, BERT, RoBERTa and XLNet embeddings, expansion of contractions was done additionally. For obligatory clauses, along with the basic preprocessing mentioned above, lemmatization was performed.

4 Baseline Models

As a first step, BERT Double, Legal BERT and Custom Legal BERT models mentioned in (L. Zheng et al., 2021) were tried for fairness classification. These models are available in HuggingFace library `https://huggingface.co/`. The scores obtained are documented in the table 2.

Model	F1 Score
BERT Double	0.76
Legal BERT	0.753
Custom Legal BERT	0.78

Table 2: BERT Variations

In CLAUDETTE (Marco Lippi et al., 2018), the authors have identified unfair clauses and the clause categories using Support Vector Machines. For fairness classification, SVM was used along with TF-IDF features and created a combination of 8 SVM's for collective classification.

For the first task (unfair clause detection), TF-IDF features were created using unigrams, bigrams and trigrams for words and parts of speech tags combined. The best model was selected using Leave One Document Out Cross Validation (LODOCV) technique. As per this technique, among 50 Terms of Service documents, each containing multiple clauses, 1 document was kept aside for testing purposes. Among the rest of the documents, cross validation was performed to select the best model by using 48 documents for training and 1 document for validation. The process was repeated 50 times and testing was done for each document. The test scores in 3 were aggregated by computing the macro average of F1 scores obtained in each iteration of LODOCV.

Model	Training	F1 Scores
SVM (words)	80-20 Split	0.72
SVM (words)	LODOCV	0.78
SVM (words & POS)	LODOCV	0.78
8 SVM (words)	LODOCV	0.73
8 SVM (words & POS)	LODOCV	0.70

Table 3: SVM with TF-IDF and LODOCV

The second task was categorizing unfairness into one of the 8 categories mentioned in 1. For this, 8 SVM classifiers (on words as well as POS tags combined) were used, each of which was trained to distinguish between unfair clauses of one category in relation to all the other categories. The F1 scores achieved can be seen in the table 3.

5 Evaluation Protocol

F1 Score (pos) : As the dataset is unevenly distributed and False Positives and False Negatives are more crucial, we have used F1 score instead of accuracy. Moreover, for fairness classification identifying unfair clauses is more important. Hence, F1 score of the positive class is considered to evaluate our models.

Accuracy : For obligations, potentially obligatory clauses to the user were manually tagged and used as a standard to measure accuracy of the model.

6 Method

6.1 Fairness Classification

The main idea was to use different embeddings with SVM and RNN based models. Because of small size of the dataset, pretrained language models were used to generate word embeddings.

1. **Generation of Embeddings**

   As a starting point, Word2Vec embeddings from a google-news-300 pretrained Word2Vec language model were used. Compared to Word2Vec, BERT embeddings are context sensitive (Jacob Devlin et al., 2019) and produce different word vectors based on different sentences. Hence, BERT embeddings using pretrained model ('bert-base-uncased') were generated. In addition to that, RoBERTa based embeddings were also created. The rationale behind using RoBERTa was that the embeddings generated by RoBERTa use dynamic masking in contrast to BERT which uses static masking. In dynamic masking RoBERTa (Yinhan Liu et al., 2019) masks different parts of the sentence in each epoch, this makes RoBERTa more robust when compared to BERT. Further, XLNet-based (Zhilin Yang et al., 2019) embeddings were generated, which improve upon BERT by introducing permutation language modeling, where all tokens are predicted in random order rather than predicting only masked tokens which is done in BERT/RoBERTa. This helps the model to learn bidirectional relationships and therefore better handle dependencies and relations between words.

2. **Simple Model - SVM**

Categories	Clauses
Train	7531 (80%)
Test	1882 (20%)

Table 4: Train Test Split

   To set a baseline for experiments, SVM with Word2Vec embeddings was used, followed by averaged word embeddings from BERT and RoBERTa. It was expected to perform better as these capture relationships and dependencies better, however it did not as seen in 5.

3. **RNN-based models**

   RNN models are good at capturing the dependencies between words in a sentence vector and can be trained on entire sentence vectors as compared to SVM. Hence various RNN based models with all the three embeddings were generated.

   The first model used basic RNN layer(s). RNN is known to have gradient exploding problem, which was encountered while training on XLNet embeddings. To mitigate this issue, gradient clipping with l2-norm and max norm=4.0 was used.

   Next, experiment used Gated Recurrent Units (GRU) and Long Short Term Memory (LSTM) which are RNN based networks equipped with memory cell to store activation values of previous words in long sentences.

   Finally, Bi-LSTM was also tried, as it is a powerful tool for modeling the sequential dependencies between words and phrases in both directions of the sequence.

   Interestingly, RNN models trained on BERT embeddings did not show any significant improvement with increased number of epochs or increased model parameters/complexity. Out of all the embeddings, the best performance noted was for RoBERTa embeddings as seen in 5.

6.2 Obligatory Clauses Detection

To identify the obligatory clauses in a ToS document, a custom word embedding model was generated after preprocessing the data. This model along with the google Word2Vec model was used to identify the keywords similar to "obligations".

ToS dataset was tagged and filtered for any occurrence of these obligatory keywords. This filtered set of clauses represents all possible obligatory clauses from the document. However, the focus was towards identifying obligatory clauses that are targeted to the user.

To identify the user-specific obligatory clauses 2 different approaches were tried, which are as under:

Approach 1:

In the first approach, a dependency parser was built for each clause to understand its grammatical structure. Parser tree was further used to mark the subject of the obligatory verbs as the entity responsible for that clause. This was done by performing BFS on the parser tree with the obligatory verb as the root. If these subject words refer to the users (such as "you", "subscriber", etc), they were marked as user-specific obligatory clauses.

Approach 2:

In this approach, the grammatical voice of the clauses i.e. active or passive was detected and also identified all the possible subjects or objects from each clause respectively. If any of the subject words refer to users, they were marked as user-specific obligatory clauses.

The first approach did not give promising results. A potential reason could be the fact that it can be really hard to accurately identify the grammatical structure of the clauses in ToS. It was noticed that the dominant verbs were not being identified accurately, thereby leading to the identification of wrong subject words. For example, if a given clause is - "if after such notice you fail to take the steps we ask of you, we'll terminate or suspend your access to the services.", the dependency parser identifies "terminate" as dominant verb and tags the organization ("we") as the subject. Hence, the second approach was used in the final model.

7 Results

7.1 Fairness Classification

The results for experiments carried out are consolidated in Table 5.

Amongst all the models, GRU with RoBERTa embeddings achieved the highest F1 score of 0.76. This is in the similar range when compared to the results of BERT model variations specified in Table 2. The final GRU model trained is shown in figure.

Model	W2V	BERT	RoBERTa	XLNet
SVM	0.49	0.35	0.29	NA
RNN	0.54	0.35	0.63	0.48
GRU	0.49	0.63	**0.76**	0.70
LSTM	0.50	0.39	0.65	0.64
Bi-LSTM	0.47	0.42	0.65	0.62

Table 5: SVM and RNN based models with various Embeddings

```
GRUModel:
    (GRU): GRU(20, 768, num_layers=1, batch_first=True)
    (classifier): Linearin_features=768, out=2)
```

Figure 1: **Model Overview**

7.2 Obligatory Clauses Detection

Accuracy for user-specific obligatory clauses detection was found to be 72%.

8 Discussion

The F1 score of 0.76 obtained with the best performing model from our experiments is in similar range with the CLAUDETTE (Marco Lippi et al., 2018) baseline models. As pretrained RoBERTa embeddings have been able to give good results, RoBERTa further finetuned on legal dataset may give promising results.

While training different models, it was observed that recall was very high, and precision was relatively low. Some of the possible reasons for this are small dataset size, and high marginal class-imbalance for unfair clauses. To mitigate class imbalance and increase dataset size, data augmentation can be performed on the rare(unfair) class. This can include applying a combination of NLP techniques like stop words removal, lemmatization and replacing words in clauses with synonyms.

If model marks all clauses as critical information, the entire document is returned to the user. This is not useful. Abstractive summarization can help mitigate this issue.

Vocabulary used in legal documents can be difficult to understand. Sentence simplification techniques can help convert sentences into layman's terms thereby making it easy for common people to read and understand the service terms.

9 Division of Labor

Various tasks involved in the proposed project and division of labour among the team members are documented in Table 6.

Task	Assignees
Try baseline benchmarks	All
Preprocessing-Obligations	Yoshitha, Aditya
Preprocessing-Fairness	Shreya, Akanksha
Entity Recognition for Obligations	Shreya, Lavina, Aditya
Fairness Classification Models	Yoshitha, Lavina, Akanksha
Documentation and Presentation	All

Table 6: Division of labor among Team members

Link to GitHub repository: `https://github.com/ShreyaPrabhu/csci-544-group18-tos-project`

References

[McDonald AM and Cranor LF2008] McDonald AM, Cranor LF. 2008. *The cost of reading privacy policies*. Isjlp 4:543.

[Marco Lippi et al.2018] Marco Lippi, Przemyslaw Palka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, Paolo Torroni. 2018. *CLAUDETTE: an Automated Detector of Potentially Unfair Clauses in Online Terms of Service*. Artificial Intelligence and Law.

[Guarino A. et al.2021] Guarino A., Lettieri N., Malandrino D. et al. 2021. *A machine learning-based approach to identify unlawful practices in online terms of service: analysis, implementation and evaluation.* Neural Comput & Applic 33, 17569–17587

[L. Zheng et al.2021] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho 2021. *When does pretraining help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset.* Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law

[Vikas and Roshann2020] Vikas Mastud and Roshan Jaiswaln. 2020. *Named Entity Recognition on legal text for secondary dataset*

[Jacob Devlin et al.2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*. https://aclanthology.org/N19-1423

[Yinhan Liu et al.2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. *RoBERTa: A Robustly Optimized BERT Pretraining Approach.*. arXiv:1907.11692 [cs.CL]

[Zhilin Yang et al.2019] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: Generalized autoregressive pretraining for language understanding.* . arXiv preprint arXiv:1906.08237.