Using Natural Language Processing to Identify Unfair Clauses in Terms and Conditions Documents

Jonathan Sears, Nick Radwin
Tulane University
jsears1@tuane.edu, nradwin@tulane.edu

Abstract

This project investigates the effectiveness of different machine learning models for sentence classification in terms and conditions documents, with a focus on identifying unfair, ambiguous, or exploitative language. The main approaches in our comparative analysis include a pretrained BERT model, a Bag of Words (BoW) logistic regression model, a Support Vector Machine (SVM), a Convolutional Neural Network (CNN), a Gradient Boosting Machine (GBM), and finally a hybrid BERT/BoW model. The primary conclusion is that while individual models offer particular strengths, the simple BoW model gives better prediction accuracy and reliability overall and is much faster than the other more complex approaches we attempted.

Introduction

No one reads Terms and Conditions, but everyone agrees to them. Every online user is faced with these incredibly dense and complex documents as if they are expected to read them in their entirety before using a digital tool or app that they want to use. In reality, we just agree to the terms and conditions to use what's behind them, leaving the user in the dark from the ambiguous, unfair, and even potentially exploitative clauses that may not be in the user's best interest.

Our project is focused on bringing these buried, questionable clauses to light so that the user can understand what he is agreeing to at a glance. As we experiment with a variety of learning models, we are building a summarization tool that takes in terms and conditions as input and generates a page that bullets a list of either ambiguous, unfair, or exploitative clauses with a confidence score for how likely the model chose those sentences correctly. We also want the tool to highlight the sentences from the original text.

Related Work

1. CLAUDETTE

https://arxiv.org/pdf/1805.01217v2.pdf

CLAUDETTE is an automated detector of potentially unfair clauses in terms and conditions, basically exactly what we are trying to build. The data they used was 50 annotated European terms of service contracts. It is worth noting they used European contracts, as different countries mean different laws, and different language used in the contracts. They used a wide variety of different ML approaches to this problem including, support vector machines, hidden markov models, LSTMs, CNNs, and different combinations of them, getting the best results (measured in F1-score) via the combined model.

2. Detecting and Explaining Unfairness in Consumer Contracts Through Memory Networks

https://link.springer.com/article/10.1007/s10506-021-09288-2

In this study, the researchers define unfair legal language in terms and conditions as language that causes "a significant imbalance in the parties' rights and obligations, to the detriment of the consumer, are deemed unfair by Consumer Law."

They note how pervasive this point out that despite substantive law in place, and despite the competence of enforcers, providers of online services still tend to use unfair and unlawful clauses in these documents."

Their system is trained on 100 labeled ToS documents and is based on splitting unfair clauses into 5 major label categories:

(i) liability exclusions and limitations

(ii) the provider’s right to unilaterally remove consumer content from the service, including in-app purchases

(iii) the provider’s right to unilaterally terminate the contract

(iv) the provider’s right to unilaterally modify the contract and/or the service

(v) arbitration on disputes arising from the contract

The unfair language/rationale pairing output was most accurate when the researchers used a memory-augmented neural network in combination with "strong supervision," meaning that they fed the expertly written rationales for specific unfair labels into their model.

3. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://aclanthology.org/2022.acl-long.297.pdf

The LexGLUE paper was written to create a benchmark for legal language understanding. It is a compilation of datasets, pretrained models and different tests and evaluation metrics performed on those datasets and models for different tasks. We think it will be useful as both a potential source and potential benchmark for our model.

4. Automatic Semantics Extraction in Law Documents

https://dl.acm.org/doi/10.1145/1165485.1165506

This study aimed to classify arguments from legislative texts using the multiclass support vector machine (MSVM) algorithm.

Stemming (representing the morphology/roots of words) improved accuracy and feature selection (restricted vocabulary) provided a work around for possible overfitting.

Overall, the study showcases MSVM's effectiveness in extracting what arguments are actually being made in legal documents, and helps us with coming up with a method for efficient legal document analysis.

5. When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset

https://arxiv.org/abs/2104.08671

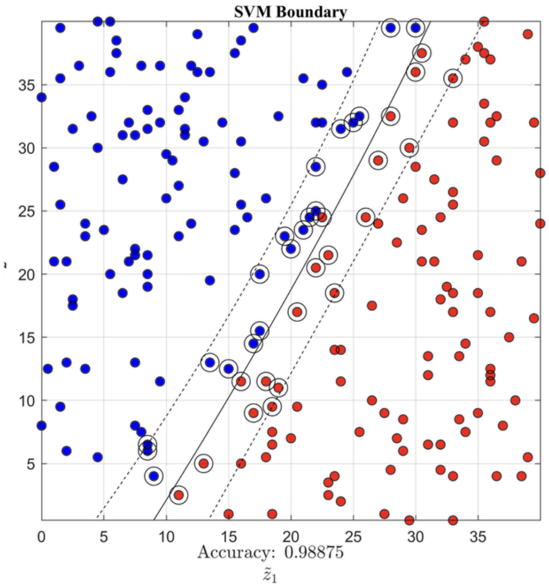
This paper presents an argument for domain specific pretraining and tests it's hypothesis on a dataset of their own creation: CaseHOLD. They compare the performance of wide domain BERT models trained separately on a corpus of legal documents (smaller, specific domain) and another model trained on wikipedia and google books (wider domain). They found that pretraining on the smaller corpus of legal documents was more effective for legal classification tasks despite there being much less data available. This leads us to believe that if we decide to use a pretrained model, we will likely use one that is domain specific to legal documents.

Since our problem is a classification problem, we think an 80-20 cross validation approach with metrics like our accuracy, precision, recall, and F1 score should suffice. However, the scale in which we perform these evaluation metrics may vary, as we may want to measure them on a document level, and on a sentence by sentence level. We can use pretrained models, as well as a simple logistic regression and naïve Bayes models as baseline metrics to compare our model too.

Approach

We used multiple different approaches to this problem. Our first instinct was to use a fine tuned BERT model, as legal language is extremely dense, so we figured the rich representation offered by BERT encodings would come in handy. To do this we utilized huggingface's transformers library along with pytorch. We used a wide variety of pretrained BERT models, including BERT-base-case, distilbert, and bert models that have already been fine tuned on legal data. Additionally, we did many types of fine tuning approaches on BERT: we tried fine tuning on the pooler output, all hidden layers, select hidden layers, and a combination of pooler output and hidden layers.

We also used a bag-of-words approach, which is a simpler representation of the text where documents are simplified into the frequencies of terms in them. We extended this representation using n-grams, stringing together multiple words and counting them as their own word. We also removed common "stop words" to get rid of filler words like "the". Finally we added stemming, which separates words from their suffixes like "-ing" and "-ed". This allows us to get a better understanding of the main idea of the sentence. On this representation we ran a wide variety of models including logistic regression, SVMs, CNN, and Gradient Boosting Machine.



The third model we tested was a Support Vector Machine for our classification task. We first learned about support vector machines in Dr. Hamm's Intro to Machine Learning Class. This model stood out because it mathematically maximizes a decision boundary between classes (we used a linear kernel because we only had 2 classes), which we thought would be useful in differentiating the subtleties in fair vs unfair language. In practice, the svm's decision boundary did not give us better results than our logistic regression, as we will explore in the Experiments section.

The fourth model we tested was a convolutional neural network. The appeal of this model is that it can automatically recognize abstract patterns within text data and learn particularly important ngrams (n=2 in our case), with the added benefit of increased efficiency once trained. Apart from the BERT model and its hybrid counterpart, the CNN is our most sophisticated experiment, but we found it still didn't perform as well as our BoW model.

The fifth was a gradient boosting machine (GBM), for a couple of reasons: GBMs employ an iterative approach to model building, enhancing model accuracy incrementally by focusing on areas where previous models underperform. This iterative improvement helps minimize overfitting, making them particularly robust. So we thought this robustness and adaptability would make it a valuable tool for identifying the nuanced patterns in legal texts, leading to accurate identification of unfair, ambiguous, or exploitative language.

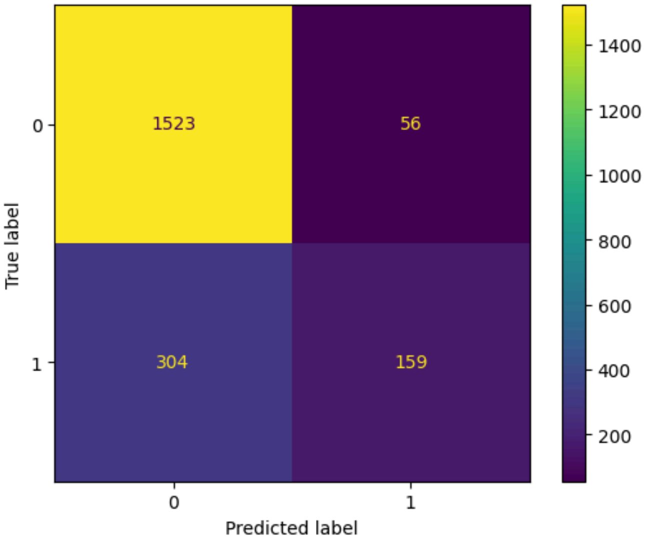
Lastly, we attempted a hybrid BERT/BoW approach with the hypothesis that a hybrid model would correctly classify sentences more often than a single model predicting on its own.

Experiments

Experiment: What kind of experiments did you do; what kind of dataset(s) you're using; what baseline method are you comparing against; and how you will evaluate your results. Report the results of your experiments in detail, including both quantitative evaluations (show numbers, figures, tables, etc) as well as qualitative evaluations (show images, example results, example errors, etc). Be sure to conduct some "stress testing" of your system using your demo interface. What type of errors does it make and why? Are there any systemic biases you notice in the system (e.g., that would affect one source of data more than others)?

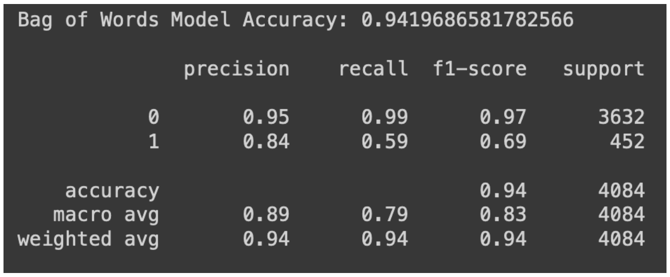
Our dataset consists of 100 labeled terms and conditions documents. The documents were labeled on a sentence level, indicating it's belonging to one of 10 classes, either fair, or one of the 9 sub categories of unfair. For simplicity, we decided to aggregate the sub-classes into one label to make our problem a binary classification problem. The distribution of classes within the dataset was extremely imbalanced, with around 92% of the sentences labeled as unfair. This distribution would prove problematic when training our models. To fix this we underfit our dataset, which meant we would train on a smaller sample size, but with an even distribution of classes.While we underfit our training data, we left our testing data as is so we could test on a distribution we would see in the real world

Initially we experimented with a BERT, however those results were underwhelming. We ran into a lot of issues when attempting to fine tune BERT. At first the BERT model was not able to pick up any differences between sentences labeled fair and sentences labeled as exploitative. After underfitting our (as described above), we still struggled to produce results with an f1-score above .5. The vast majority of our training runs resulted in something like the figure to the right. While on some occasions we were able to get higher precision, we were never able to achieve the balance between precision and recall needed for a good f1-score. Our theory as to why BERT is not sufficient is that since the semantics of legal language are similar regardless of fair or unfair language, BERT may struggle to separate the two because their representations will be so similar.

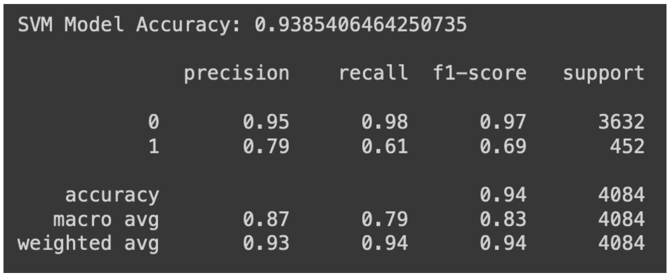


We encountered and overcame many errors in our exploration of different model types.

For the BoW model, the logistic regression approach was straightforward, but we encountered an issue where the confusion matrix function in scikit-learn required the inputs (labels_test and labels_pred) to have the same data type. Correcting this required us to convert labels_test to integers. We went on to encounter similar data-type mismatch errors in other models, which we corrected by converting all types to “int.” We used sklearn;s SVM model, and we used the preprocessing we performed from the BoW model to see if fitting on the SVM prediction would give better results (it did not, see classification reports in the figures. We saw

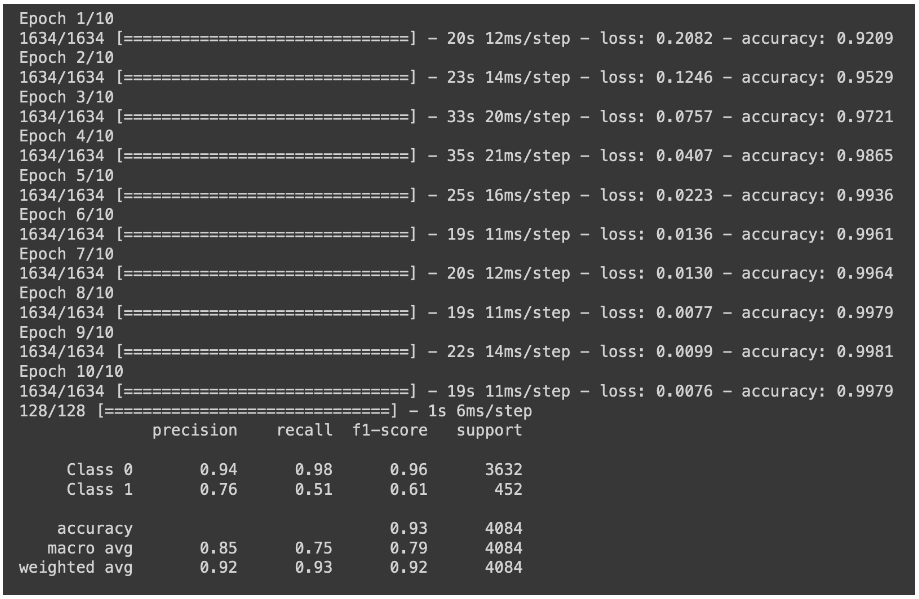


```
Bag of Words Model Accuracy: 0.9419686581782566

              precision    recall  f1-score   support

           0       0.95      0.99      0.97      3632
           1       0.84      0.59      0.69       452

    accuracy                           0.94      4084
   macro avg       0.89      0.79      0.83      4084
weighted avg       0.94      0.94      0.94      4084
```



```
SVM Model Accuracy: 0.9385406464250735

              precision    recall  f1-score   support

           0       0.95      0.98      0.97      3632
           1       0.79      0.61      0.69       452

    accuracy                           0.94      4084
   macro avg       0.87      0.79      0.83      4084
weighted avg       0.93      0.94      0.94      4084
```

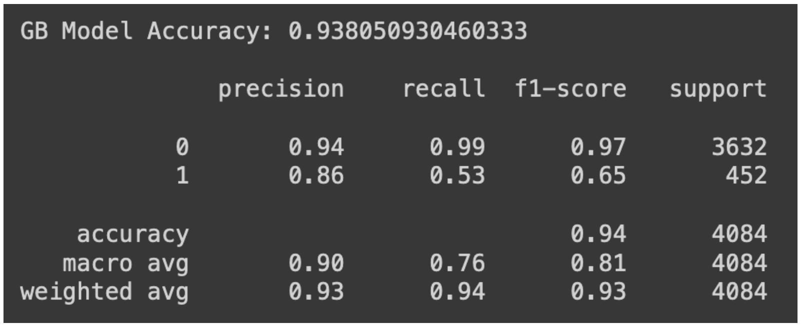
The CNN model was uniquely challenging, since we needed to format the input data as padded sequences of tokens which was definitely a learning curve. There were also issues when attempting to integrate the CNN into our scikit-learn workflow, which doesn't natively support deep learning models. Aside from BERT and the hybrid model, this model took forever to implement because it involves a large corpus of tensorflow libraries and 4 distinct layers to make a prediction–the representation layer to compute the embeddings, the convolution layer to extract features and recognize patterns in the unfair sentences, the pooling layer to distill these features into the most important ones, and the classification layer to flatten these important features into a single vector. Overall its results were slightly worse than our more simple Bag of Words (BoW) model based on the precision, recall, and f1 scores for Class 1 (unfair language). These results surprised us because although convolutional neural networks are usually used for image processing, they are supposed to be



```
Epoch 1/10
1634/1634 [==============================] - 20s 12ms/step - loss: 0.2082 - accuracy: 0.9209
Epoch 2/10
1634/1634 [==============================] - 23s 14ms/step - loss: 0.1246 - accuracy: 0.9529
Epoch 3/10
1634/1634 [==============================] - 33s 20ms/step - loss: 0.0757 - accuracy: 0.9721
Epoch 4/10
1634/1634 [==============================] - 35s 21ms/step - loss: 0.0407 - accuracy: 0.9865
Epoch 5/10
1634/1634 [==============================] - 25s 16ms/step - loss: 0.0223 - accuracy: 0.9936
Epoch 6/10
1634/1634 [==============================] - 19s 11ms/step - loss: 0.0136 - accuracy: 0.9961
Epoch 7/10
1634/1634 [==============================] - 20s 12ms/step - loss: 0.0130 - accuracy: 0.9964
Epoch 8/10
1634/1634 [==============================] - 19s 11ms/step - loss: 0.0077 - accuracy: 0.9979
Epoch 9/10
1634/1634 [==============================] - 22s 14ms/step - loss: 0.0099 - accuracy: 0.9981
Epoch 10/10
1634/1634 [==============================] - 19s 11ms/step - loss: 0.0076 - accuracy: 0.9979
128/128 [==============================] - 1s 6ms/step
              precision    recall  f1-score   support

     Class 0       0.94      0.98      0.96      3632
     Class 1       0.76      0.51      0.61       452

    accuracy                           0.93      4084
   macro avg       0.85      0.75      0.79      4084
weighted avg       0.92      0.93      0.92      4084
```

good at adapting to detect local patterns in language and its abstract aspects, in our case, aspects of language. But as we see above, after 10 epochs the results are not as good as our BoW model. The CNN's lower performance suggests that the local patterns of language in terms and conditions are not as important as the specific words used in terms of classifying as unfair or not.

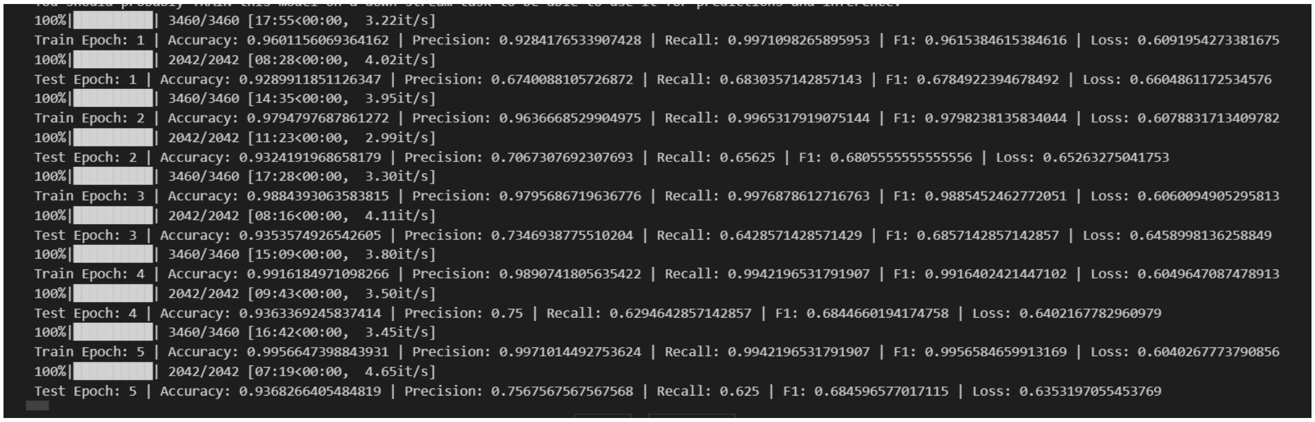
As we learned in Intro to Machine Learning, configuring our GBM involved careful tuning of parameters so it would serve for text classification. We also were getting horrendous results until we started using sklearn’s TF-IDF tool, which weights words based on their frequency across all documents, giving less importance to commonly occurring words and more to rare words that provide specific insights into document content, whereas CountVectorizer (which we used in our BoW model) simply counts the number of times each word appears in a document. We again had to tinker with data type mismatch problems, and our parameter tuning such as tree depth and learning rate required iterative testing and validation to optimize performance.



```
GB Model Accuracy: 0.938050930460333

              precision    recall  f1-score   support

           0       0.94      0.99      0.97      3632
           1       0.86      0.53      0.65       452

    accuracy                           0.94      4084
   macro avg       0.90      0.76      0.81      4084
weighted avg       0.93      0.94      0.93      4084
```

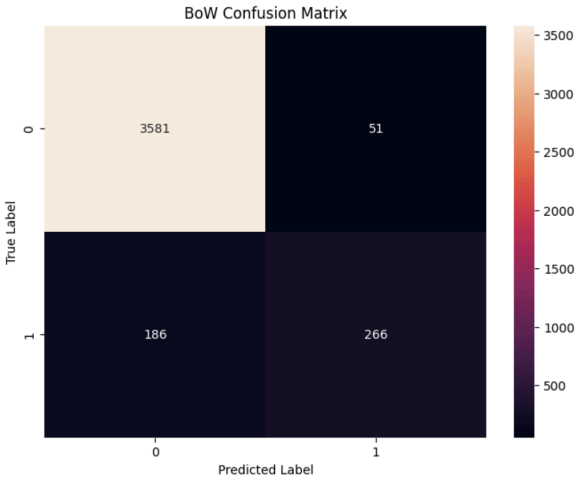
In the end, we achieved no improvements over the logistic regression in our BoW model despite all the fancy tricks of the other models.

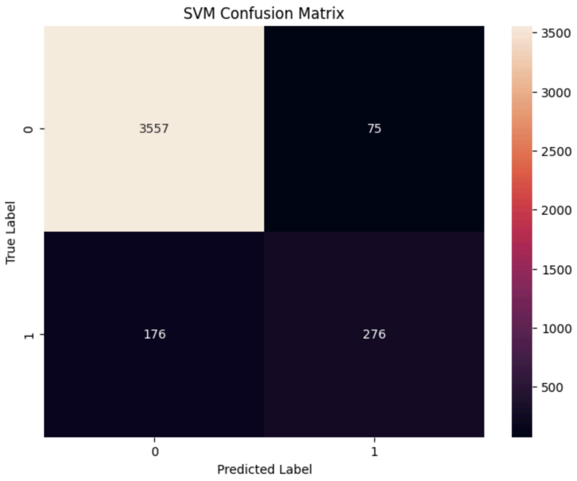
Finally we tried to use a hybrid model with BERT and the logistic regression variant of the bag of words model. We did this by creating a traditional fine tune BERT setup, but instead of going directly from bert embeddings to predictions we passed them through a smaller linear layer to get a 1x2 vector. We also passed the output of the bag of words model through the forward function of the neural network and then concatenated that (a 1x2 vector) and the 1x2 vector learned from the BERT model into one 1x4 vector. We then passed that vector through the final classifier linear layer to get our final output. Our hope in doing this is that the final classification layers will properly weight the output from both BERT and bag of words. Since our bag of words model had high precision but low recall, and the BERT models had low precision and high recall, our hope was that by combining the two we will get the best of both worlds. This ended up being somewhat the case, as the hybrid model was able to achieve a solid balance between precision and recall. However, its f1-scores were still not able to surpass that of the original bag of words model, plateauing at around .685. Despite a slightly lower f1 score, we still claim the hybrid model to be somewhat successful, as it did manage to achieve a more even distribution of precision and recall than even the Logistic Regression bag of words model.

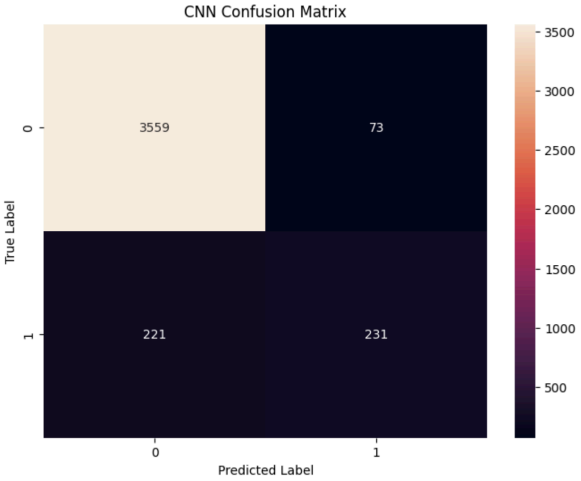


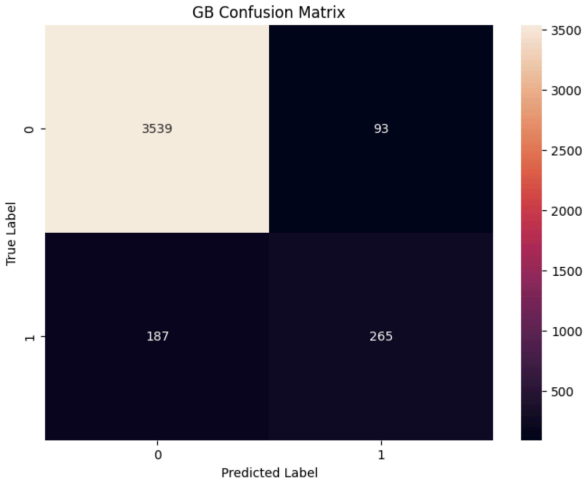
```
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
100%|██████████| 3460/3460 [17:55<00:00,  3.22it/s]
Train Epoch: 1 | Accuracy: 0.9601156069364162 | Precision: 0.9284176533907428 | Recall: 0.9971098265895953 | F1: 0.9615384615384616 | Loss: 0.6091954273381675
100%|██████████| 2042/2042 [08:28<00:00,  4.02it/s]
Test Epoch: 1 | Accuracy: 0.9289911851126347 | Precision: 0.6740088105726872 | Recall: 0.6830357142857143 | F1: 0.6784922394678492 | Loss: 0.6604861172534576
100%|██████████| 3460/3460 [14:35<00:00,  3.95it/s]
Train Epoch: 2 | Accuracy: 0.9794797687861272 | Precision: 0.9636668529904975 | Recall: 0.9965317919075144 | F1: 0.9798238135834044 | Loss: 0.6078831713409782
100%|██████████| 2042/2042 [11:23<00:00,  2.99it/s]
Test Epoch: 2 | Accuracy: 0.9324191968658179 | Precision: 0.7067307692307693 | Recall: 0.65625 | F1: 0.6805555555555556 | Loss: 0.65263275041753
100%|██████████| 3460/3460 [17:28<00:00,  3.30it/s]
Train Epoch: 3 | Accuracy: 0.9884393063583815 | Precision: 0.9795686719636776 | Recall: 0.9976878612716763 | F1: 0.9885452462772051 | Loss: 0.6060094905295813
100%|██████████| 2042/2042 [08:16<00:00,  4.11it/s]
Test Epoch: 3 | Accuracy: 0.9353574926542605 | Precision: 0.7346938775510204 | Recall: 0.6428571428571429 | F1: 0.6857142857142857 | Loss: 0.645899813625849
100%|██████████| 3460/3460 [15:09<00:00,  3.80it/s]
Train Epoch: 4 | Accuracy: 0.9916184971098266 | Precision: 0.9890741805635422 | Recall: 0.9942196531791907 | F1: 0.9916402421447102 | Loss: 0.6049647087478913
100%|██████████| 2042/2042 [09:43<00:00,  3.50it/s]
Test Epoch: 4 | Accuracy: 0.9363369245837414 | Precision: 0.75 | Recall: 0.6294642857142857 | F1: 0.6844660194174758 | Loss: 0.6402167782960979
100%|██████████| 3460/3460 [16:42<00:00,  3.45it/s]
Train Epoch: 5 | Accuracy: 0.9956647398843931 | Precision: 0.9971014492753624 | Recall: 0.9942196531791907 | F1: 0.9956584659913169 | Loss: 0.6040267773790856
100%|██████████| 2042/2042 [07:19<00:00,  4.65it/s]
Test Epoch: 5 | Accuracy: 0.9368266405484819 | Precision: 0.7567567567567568 | Recall: 0.625 | F1: 0.684596577017115 | Loss: 0.6353197055453769
```

Hybrid Model training results





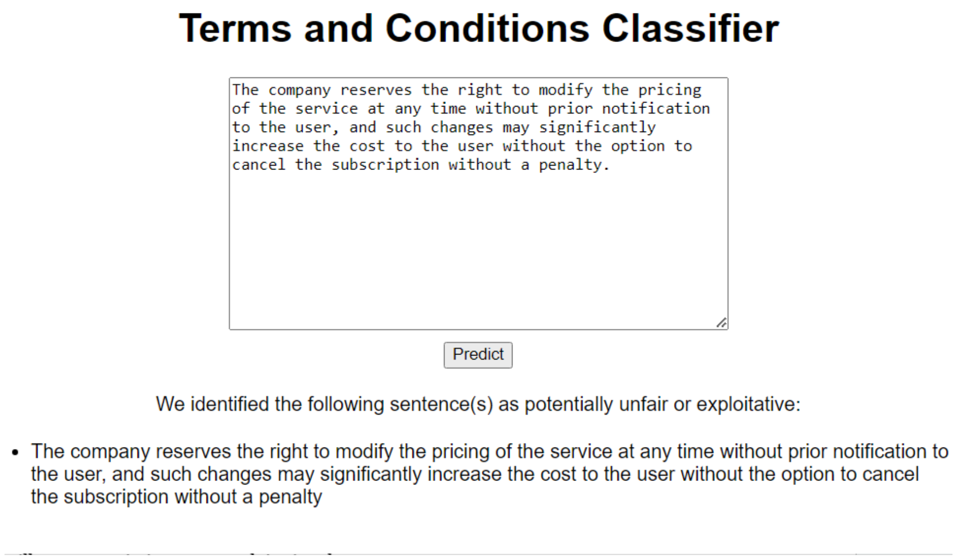




Demo

We ended up using our logistic regression model trained on the bag of words dataset as the model in our demo, as it was our best performing model from our experiments (highest precision and f1-score).

While the model can Identify some potentially unfair sentences:



Terms and Conditions Classifier

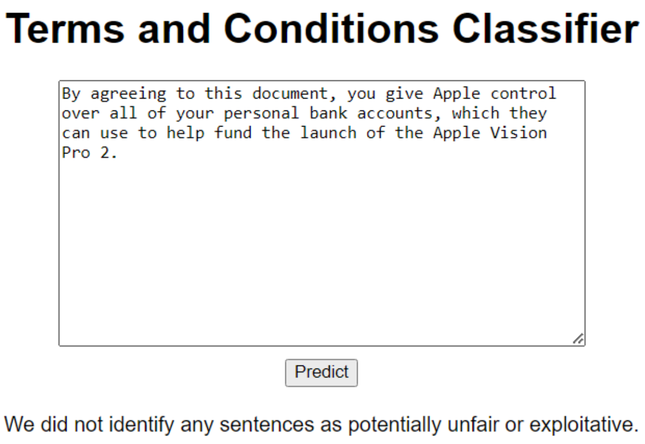
```
The company reserves the right to modify the pricing
of the service at any time without prior notification
to the user, and such changes may significantly
increase the cost to the user without the option to
cancel the subscription without a penalty.
```

We identified the following sentence(s) as potentially unfair or exploitative:

- The company reserves the right to modify the pricing of the service at any time without prior notification to the user, and such changes may significantly increase the cost to the user without the option to cancel the subscription without a penalty

It is still prone to missing some exploitative clauses.

Including extremely really blatant ones...



Terms and Conditions Classifier

```
By agreeing to this document, you give Apple control
over all of your personal bank accounts, which they
can use to help fund the launch of the Apple Vision
Pro 2.
```

We did not identify any sentences as potentially unfair or exploitative.

When we passed the entire YouTube Terms and Conditions document into our demo we got the following output:

We identified the following sentence(s) as potentially unfair or exploitative:

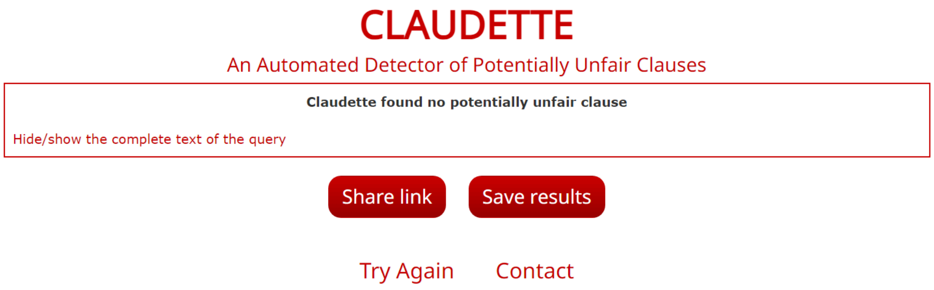
- Applicable Terms Your use of the Service is subject to these terms, the YouTube Community Guidelines and the Policy, Safety and Copyright Policies which may be updated from time to time (together, this "Agreement")
- As part of this continual evolution, we may make modifications or changes (to all or part of the Service) such as adding or removing features and functionalities, offering new digital content or services or discontinuing old ones
- Removing Your Content You may remove your Content from the Service at any time
- Removal of Content By YouTube If any of your Content (1) is in breach of this Agreement or (2) may cause harm to YouTube, our users, or third parties, we reserve the right to remove or take down some or all of such Content in our discretion
- Violation of this prohibition is a material breach of this Agreement and Google reserves the right to terminate your Google account or your access to all or part of the Service
- YouTube's policies provide for the termination, in appropriate circumstances, of repeat infringers' access to the Service
- Account Suspension & Termination Terminations by You You may stop using the Service at any time
- Terminations and Suspensions by YouTube YouTube reserves the right to suspend or terminate your Google account or your access to all or part of the Service if (a) you materially or repeatedly breach this Agreement; (b) we are required to do so to comply with a legal requirement or a court order; or (c) we reasonably believe that there has been conduct that creates (or could create) liability or harm to any user, other third party, YouTube or our Affiliates
- Effect of Account Suspension or Termination If your Google account is terminated or your access to the Service is restricted, you may continue using certain aspects of the Service (such as viewing only) without an account, and this Agreement will continue to apply to such use
- Limitation of Liability EXCEPT AS REQUIRED BY APPLICABLE LAW, YOUTUBE, ITS AFFILIATES, OFFICERS, DIRECTORS, EMPLOYEES AND AGENTS WILL NOT BE RESPONSIBLE FOR ANY LOSS OF PROFITS, REVENUES, BUSINESS OPPORTUNITIES, GOODWILL, OR ANTICIPATED SAVINGS; LOSS OR CORRUPTION OF DATA; INDIRECT OR CONSEQUENTIAL LOSS; PUNITIVE DAMAGES CAUSED BY: ERRORS, MISTAKES, OR INACCURACIES ON THE SERVICE; PERSONAL INJURY OR PROPERTY DAMAGE RESULTING FROM YOUR USE OF THE SERVICE; ANY UNAUTHORIZED ACCESS TO OR USE OF THE SERVICE; ANY INTERRUPTION OR CESSATION OF THE SERVICE; ANY VIRUSES OR MALICIOUS CODE TRANSMITTED TO OR THROUGH THE SERVICE BY ANY THIRD PARTY; ANY CONTENT WHETHER SUBMITTED BY A USER OR YOUTUBE, INCLUDING YOUR USE OF CONTENT; AND/OR THE REMOVAL OR UNAVAILABILITY OF ANY CONTENT
- YOUTUBE AND ITS AFFILIATES' TOTAL LIABILITY FOR ANY CLAIMS ARISING FROM OR RELATING TO THE SERVICE IS LIMITED TO THE GREATER OF: (A) THE AMOUNT OF REVENUE THAT YOUTUBE HAS PAID TO YOU FROM YOUR USE OF

THE SERVICE IN THE 12 MONTHS BEFORE THE DATE OF YOUR NOTICE, IN WRITING TO YOUTUBE, OF THE CLAIM; AND (B) USD $500

- Indemnity To the extent permitted by applicable law, you agree to defend, indemnify and hold harmless YouTube, its Affiliates, officers, directors, employees and agents, from and against any and all claims, damages, obligations, losses, liabilities, costs or debt, and expenses (including but not limited to attorney's fees) arising from: (i) your use of and access to the Service; (ii) your violation of any term of this Agreement; (iii) your violation of any third party right, including without limitation any copyright, property, or privacy right; or (iv) any claim that your Content caused damage to a third party
- About this Agreement Changing this Agreement We may change this Agreement, for example, (1) to reflect changes to our Service or how we do business - for example, when we add new products or features or remove old ones, (2) for legal, regulatory, or security reasons, or (3) to prevent abuse or harm
- If you don't agree to the new terms, you should remove any Content you uploaded and stop using the Service
- Governing Law All claims arising out of or relating to these terms or the Service will be governed by California law, except California's conflict of laws rules, and will be litigated exclusively in the federal or state courts of Santa Clara County, California, USA
- You and YouTube consent to personal jurisdiction in those courts

In our non-expert, non lawyer opinions, the highlighted terms seem pretty unfair to us.

Compared with the output of Claudette:



CLAUDETTE

An Automated Detector of Potentially Unfair Clauses

Claudette found no potentially unfair clause

While our classifier may not be perfect, or accurate, we are still really proud of the work we did and the valuable skills we learned along the way.

Conclusion

In conclusion our simplest model, the BoW logistic regression was the most effective overall despite other models like the SVM having a slight improvement in correctly classifying sentences as unfair, ambiguous, or exploitative. Furthermore, no model was better than the BoW at classifying fair sentences as the BoW model. Although, the complex nature of working on and tinkering with the CNN, BERT, and Hybrid BERT/BoW models taught us the most about how best to approach this classification task in general and natural language processing as a whole.

Looking into the future, we still believe that better results can be achieved with an ensemble model, as proven by the original CLAUDETTE paper. That being said, it is hard to beat the effectiveness and efficiency of a BoW logistic regression model.

References

Konstantinou, C. (n.d.). *SVM Decision Boundary*. https://www.researchgate.net/figure/SVM-decision-boundary_fig1_340331352.

NLTK Project. "nltk.corpus.stopwords - NLTK 3.5 documentation." *NLTK.org*, Natural Language Toolkit, https://www.nltk.org/modules/nltk/corpus/stopwords.html.

NLTK Project. "nltk.stem.PorterStemmer - NLTK 3.5 documentation." *NLTK.org*, Natural Language Toolkit, https://www.nltk.org/modules/nltk/stem/porter.html.

Scikit-Learn Development Team. "SVC - scikit-learn 0.24.1 documentation." *Scikit-Learn.org*, Scikit-Learn, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

Scikit-Learn Development Team. "TfidfVectorizer - scikit-learn 0.24.1 documentation." *Scikit-Learn.org*, Scikit-Learn, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.