Project Report: NLP on Privacy policies

Siddharth Tankariya
siddharth.tankariya@knights.ucf.edu

Pranav Bhasin
pranav.bhasin@knights.ucf.edu

Vinay Ambre
vinay.ambre@knights.ucf.edu

Kesar TN
kesar@knights.ucf.edu

1. ABSTRACT

Privacy documents are an integral part of businesses that allows user to understand the use of user information for business purposes. These privacy documents are lengthy, containing jargon which is difficult for the users to understand. We try to build a classifier that classifies legal documents in various categories which helps user understand in which category the legal document focuses on. We implement this classifier using various classifier architectures like Naive Bayes, CNN, BiGRU and BERT. We also perform extractive summarization on legal documents with the help of topic modelling. This provides a short summary of the legal documents.

2. INTRODUCTION

In today's data driven day and age user data has become a driving force behind many major businesses. In order to use the user data companies make consumers accept their privacy policies. The privacy policies are myriad with lots of legal jargon which discourages users to read them carefully. We task our classification models to classify the documents into 9 different categories such as Data security, Data tracking etc. The dataset has 455 instances and is trained on 5 different NLP models, with rudimentary algorithms such as Naive Bayes and CNN along with more advanced architectures such as LSTM, Bi-directional GRU etc. We also try to understand the data even further by extractive summarization of these training instances using Topic modeling implementing the LDA algorithm.

2.1 Motivation

Privacy policies are long legal procedures explaining the users data collection and practices for a particular website. However, since the policies are complex to understand, the users prefer to skip it and agree to the terms. The legal jargon is unclear as to whether it the policies are addressing the crucial aspects such as Security, Data retention etc. The manual annotation is time-consuming and is bottlenecked due to the amount of information provided in a single document. Hence, we make an attempt to understand these policies as to which industry it is trying to address. This model will further help automate the annotation process of these lengthy documents. The privacy categories are thoroughly analyzed by NLP techniques seen in our project.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

2.2 Problem Statement

Our System will try to label these privacy policies into 9 different categories which are *First Party Collection, Third Party Sharing, User choice, User Access Edit and Deletion, Data Retention, Do not Track, Policy Change, Data Security, and International and Specific Audience.* It is a multi-labeled system which means it will also capture the correlation amongst the various categories.[1] This will allow the users to better understand the privacy policies and make a decision whether to read it or not. Moreover, We will summarize the long policy documents according to the topics they belong. This will help the user gain more information about the policy without actually having to read the whole document.

3. RELATED WORK

The goal of Multi-label classification is to assign a set of relavant labels for a single instance.

Various algorithms are used for multilabel classification problems. The binary relevance method is an ensemble of single-label binary classifiers trained independently on each class. The label powerset approach maps each combination of labels into a single label using which the model is trained on. While the classifier-chain model is a chain of binary classifiers, linked together. In our project, we will look into the most efficient algorithm for our problem statement and try to inculcate latest transfer learning models for the same.

4. DATASET

Our dataset consists of 455 privacy policies belonging to 9 categories. Every policy document belong to a subset of these 9 categories according to the document content. The dataset consisted of a text document containing the privacy policies and an excel file containing the categories belonging to every document. We performed data exploration on these categories to find various relationships between these categories and documents. Initially we pre-processed the data using a pipeline to clean the data and make it consistent in format. The documents were passed through regex filters, stopwords removal, lowercasing and then tokenized. These

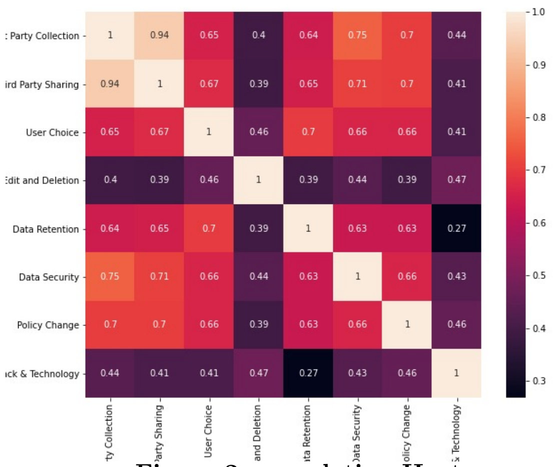

Figure 2: correlation Heatmap

Figure 3: Wordnet for most common words

filtered text documents were then used for visualizing trends in categories and text documents. The distribution of categories over the entire text corpus can be seen as :-

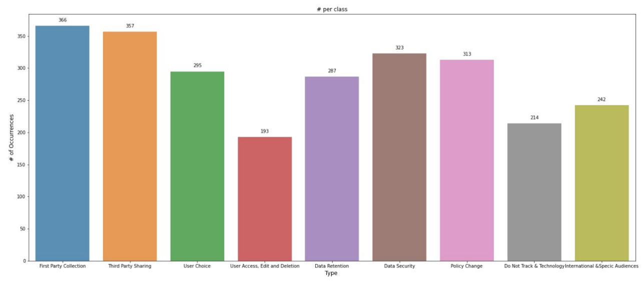

Figure 1: Distribution of categories

The correlation heat map between the policy categories can be visualised as :-

As we can see, there is a high correlation between category 'data security' and 'third party sharing'. There is even a low correlation of the category 'User access, edit and deletion' with the other categories in the dataset. Moreover, the number of words that contribute to the categories in the dataset were also discovered using the TF IDF score. This can be visualised as:-

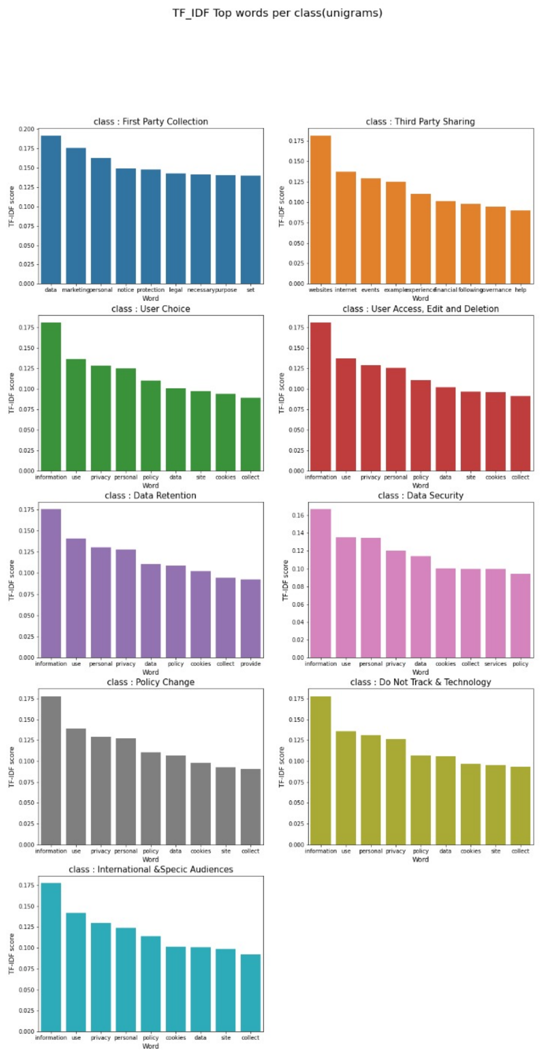

Figure 4: distribution of data

5. IMPLEMENTATION

We implemented 5 different models on the policy documents for predicting the multi-label classes. The models are Naive Bayes, LSTM, Convolutional Neural Network, Bi-GRU and BERT. The last four were neural network based approach in which GLOVE word embedding[2] were used. The dataset was split in training and testing datasets for simpler evaluations of each model. We did not retrain the GLOVE word representation since these word vectors are trained on huge corpus which gives better representation than re-training on our dataset. These word embeddings were passed in the neural network using the Embedding layer in keras. The model and results can be seen on the project website **Multi-label-Classifier-on-Privacy-Policies**

5.1 Pipeline

The pipeline for the entire system can be viewed as in the figure. The data pre-processing stage consisted of removing special characters from the policy documents, removing stopwords, seperating each document and then storing them in a dataframe for furthur computations. For, classification purposes we split the dataframe into train and test dataframes and then passed them through a tokenizer. The tokens are then passed to the input layers of the respective classifiers. For summarization, the dataframe is passed through a count vectorizer before passing it through a LDA model to find topics.

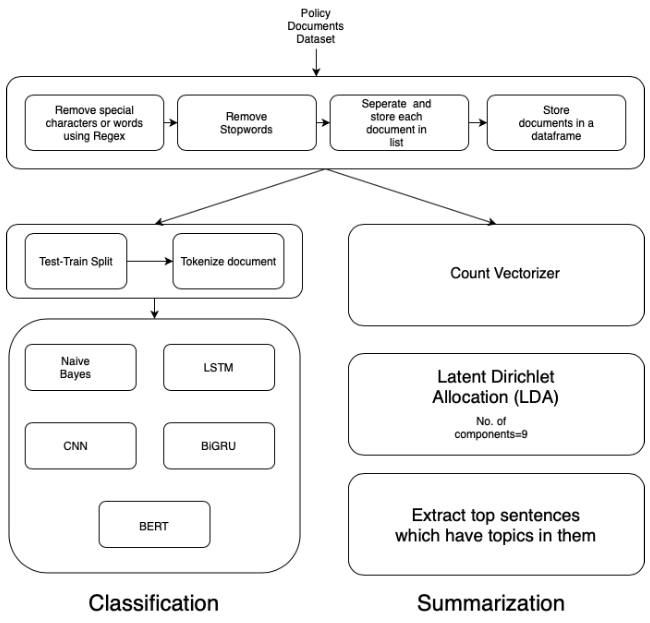
Policy Documents Dataset
Remove special characters or words using Regex
Remove Stopwords
Seperate and store each document in list
Store documents in a dataframe
Test-Train Split
Tokenize document
Naive Bayes
LSTM
CNN
BiGRU
BERT
Count Vectorizer
Latent Dirichlet Allocation (LDA)
No. of components=9
Extract top sentences which have topics in them
Classification
Summarization

Figure 5: Project Pipeline

5.2 Naive-Bayes

The first model is Multinomial Naive Bayes approach. Naive Bayes uses a bag of words representation of documents for classification. Likelihoods and priors are calculated by knowing the occurences of the words in the documents. The simple equation for the Naive bayes can be given as

$$p(f_1, f_2, ...f_n|c) = \prod_{i=a}^{n} f(f_i|c)$$

where f is the features of the data and c is the class. Hence, it is a generative model. The count vector matrix was calculated for this purpose and passed to the model for training.

The model pipeline can be seen below:

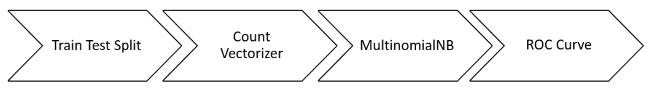

Figure 6: naive bayes pipeline

Since this is a Multi-Label classification, We run the Multinomial Naive bayes equation 9 different times to get the 9 different predictions. The ROC curve for each category was calculated which can be visualised as:-

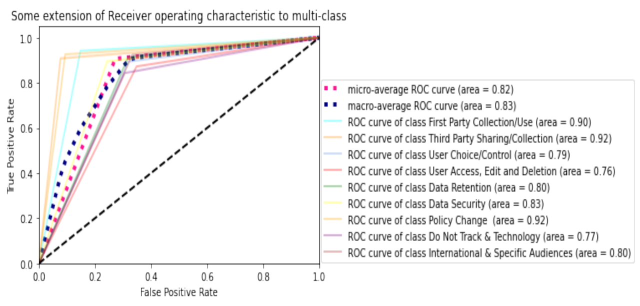

Figure 7: naive bayes results

5.3 CNN Model

Another model that we tried classifying the privacy policies on was the CNN model. Here we used convolution layers with different filter sizes. We use convolution layers in NLP just like we use convolution layers in Images. Here, instead of the performing convolution over a matrix of image pixels we perform convolution over the matrix of sentence. This matrix contains rows representing the word embeddings or also sometimes character embeddings. With the help of this features are extracted from the sentences which help in learning the model.

The architecture of CNN that we used contains 5 convolution layers with different sizes each. Each convolution layer is followed by a max pooling layer[3]. The output from these 5 layers is concatenated to be passed further. This extracts the features in the sentences and reduces the dimensions to be processed. After the concatenation, the features are passed to an alternate dropout and dense layers. The final output layer has a sigmoid activation with 9 nodes. These 9 nodes are the 9 categories present in the dataset. The sigmoid function outputs a probability distribution between 0 and 1. The input representations are padded at the end by 0 before passing to the network.

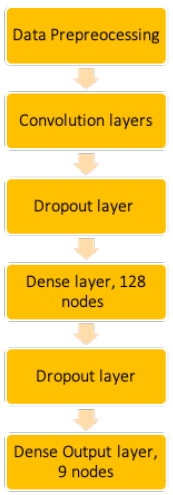

Figure 8: CNN pipeline

As we can see from the ROC curve for the CNN classification model. The ROC curve for all the target classes is represented here on the graph. The CNN model doesn't perform as good as for Naïve Bayes or LSTM.

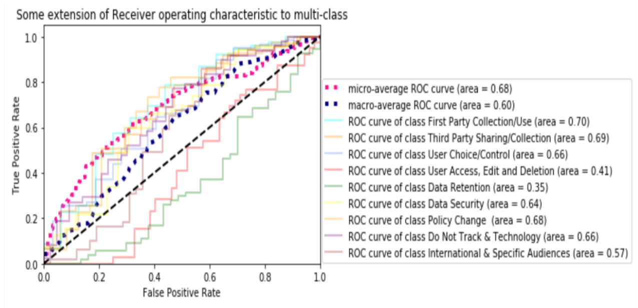

Figure 9: CNN results

Since, the input sequence must all have the same size, the sentence representations are padded with 0 before passing them to the input layer.

5.4 LSTM

Long Short Term Memory networks help capture the semantic representations of the long term dependencies. It is a special case of Recurrent Neural network. It helps us address the vanishing gradient problem that arises in RNNs [4]. After the Train-Test split, we pass the data through a Tokenizer function to obtain the token IDs for the unique tokens in our data. We then use the GloVe embedding model to obtain the embedding matrix. It is an unsupervised learning algorithm used to obtain the vector representation of the words. The Glove 6B model is trained on Wikipedia 2014 + Gigaword 5 datasets which has 400k vocab, 6B tokens with 100 dimensions, trained on the aggregated word-to-word co-occurrence statistics from the data. The resulting representation provides a linear substructure of the word-vector space [Glove]. These word embeddings are fed into our LSTM module with 128 timesteps. This extracts the neccessary features which is then fed into 9 different dense layers, each representing a single class from our problem statement. We can see the pipeline diagram for this particular model below.

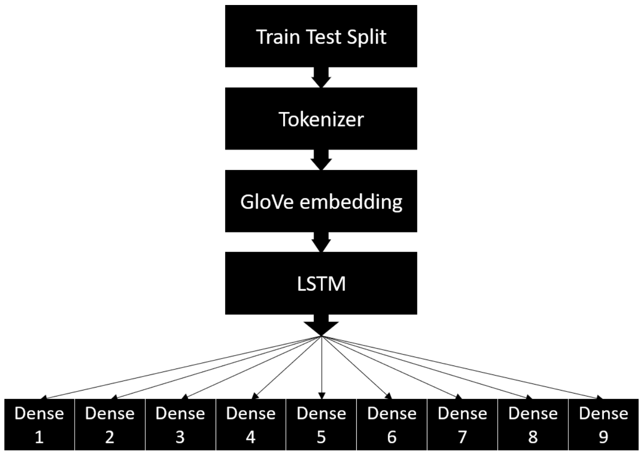

Figure 10: LSTM pipeline

We use Binary cross-entropy loss functions for each of these dense layers providing us result either as 1 or 0 for all 9 classes. This model provided us with the following ROC curve.

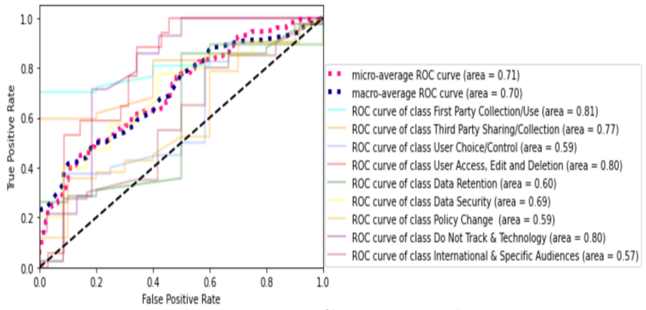

Figure 11: LSTM Results

5.5 BiGRU

The next model that we used for the classification was Bidirectional GRU(Gated Recurrent Unit). In this the sentence is not only passed through the network from left to right like in LSTM but also passed from right to left. This helps in solving the issues of long dependencies in the words of the sentence. The hidden state used in BiGRU captures the weights of the sentence when it is passed in both the directions.

In our BiGRU architecture, we have an embedding layer which is followed by a spatial dropout layer. The maximum length is set to 8000. It then follows a Bidirectional GRU layer. The output from this layer is then passed through a convolution layer. The output from the convolution layer is passed through two pooling layer. One of them is max pooling layer and the other is average pooling layer. The output from those pooling layers is concatenated and passed to a dense output layer. The output layer has a sigmoid activation just like in CNN architecture which has 9 nodes. These 9 nodes represent the target classes in the dataset. The sigmoid activation outputs a value between 0 and 1.

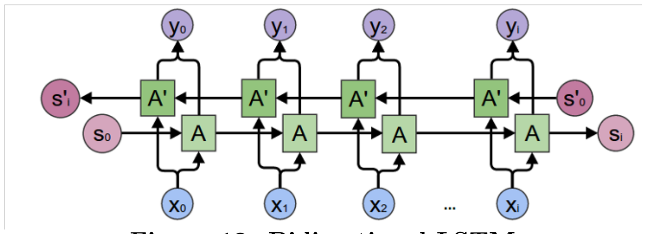

Figure 12: Bidirectional LSTM

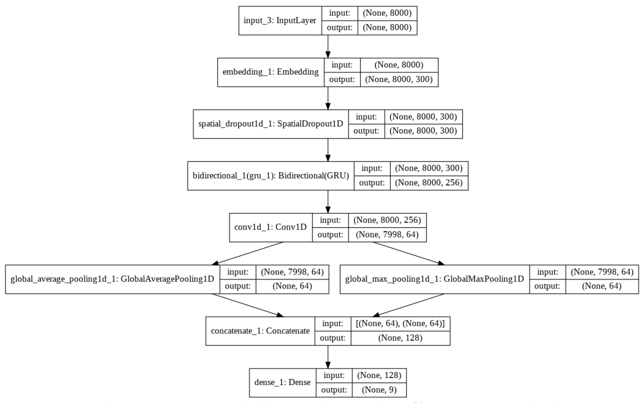

Figure 13: Bidirectional LSTM Model

As we can see from the ROC curve for the Bidirectional GRU it performs better than the CNN model as we can see the area under the curve for all the classes.

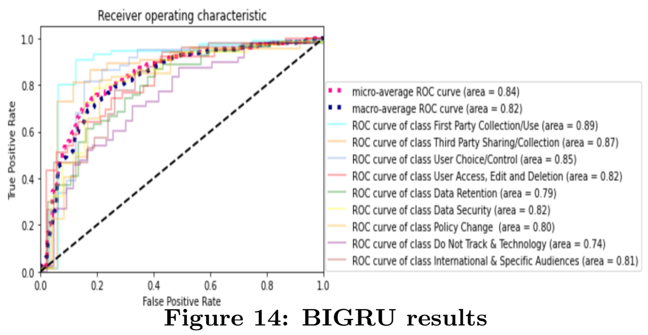

Figure 14: BIGRU results

5.6 Bert Based - classifier Model

The next model that we implement is using a Bi-directional Language Model called BERT. BERT is released in 2018 and become hugely popular as it was first to achieve language modeling in a bidirectional fashion. It achieved State of the art in many NLP tasks. We used BERT as it has been trained on a huge corpus and we expect it to perform very well for a classification task. Since BERT takes input a maximum input of 512 tokens we truncate all documents to a maximum of 512 tokens. We have a dropout layer and a linear classifier layer which is used for prediction. The results can be seen in the figure below :-

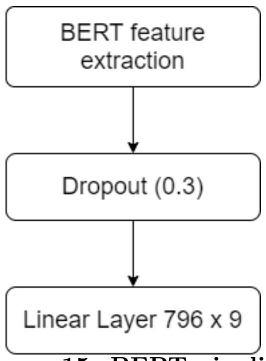

Figure 15: BERT pipeline

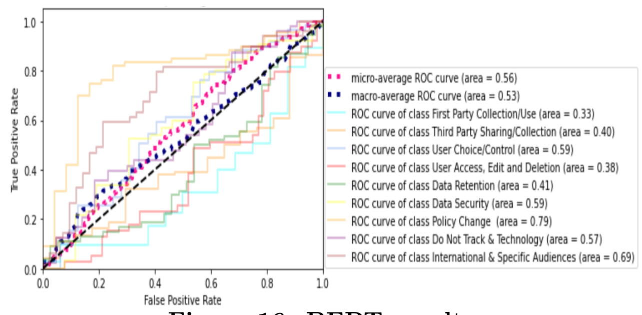

Figure 16: BERT results

5.7 Extractive Summarization

In this section we will discuss the Extractive Summarization discussed

5.7.1 *Topic Modeling:*

Topic modeling is the process of identifying topics in a set of documents. In our case we use it to generate summaries. Many algorithm exist for topic modeling but we choose to go for **Latent Dirichlet allocation(LDA)** [5]

LDA uses the bag-of-word technique and looks at all the word counts to generate features. LDA works on the basis of an assumption that all documents are generated with a set of topics and then the words in that topic are generated.

We try to use this principle for generating our summarizations. We generate 9 topics and generate summaries for each topic by choosing the top 4 sentences with the most words from that topic.

We build a dashboard to visualize the LDA below.

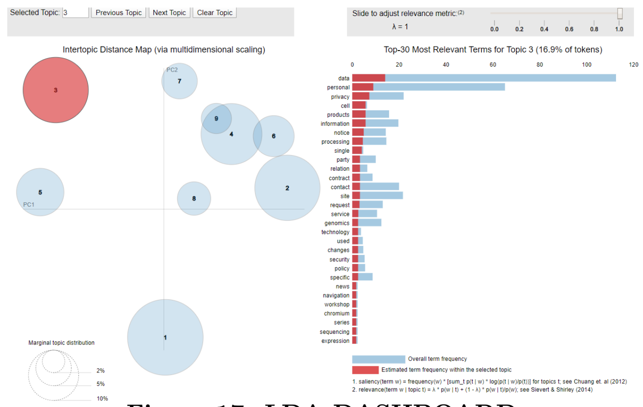
Selected Topic: 3
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:
λ = 1
Intertopic Distance Map (via multidimensional scaling)
Top-30 Most Relevant Terms for Topic 3 (16.9% of tokens)
Marginal topic distribution
Overall term frequency
Estimated term frequency within the selected topic

Figure 17: LDA DASHBOARD

6. EVALUATION

Our dataset consist of multi-label samples, wherein the text corpus can belong to a subset of classes rather than only one class. Hence, this makes the evaluation metrics like accuracy, precision, recall, etc redundant. This is because it will not consider the partially correct predictions and thus misleading to wrong evaluation metrics.

To overcome this, evaluation metrics like micro averaging and macro averaging are used. In Micro averaging we sum up the individual true positives(TP), false positives(FP), and false negatives(FN) of the system for different classes and the apply them.

$$MicroAveragingPrecision = \frac{\sum_{c_i \epsilon C} TPs(c_i)}{\sum_{c_i \epsilon C} TPs(c_i) + FPs(c_i)}$$

$$MicroAveragingRecall = \frac{\sum_{c_i \epsilon C} TPs(c_i)}{\sum_{c_i \epsilon C} TPs(c_i) + FNs(c_i)}$$

In macro averaging, we take the average of precision and recall for different classes in the dataset and take the average.

$$MacroAveragingPrecision = \frac{\sum_{c_i \epsilon C} Prc(D, c_i)}{|c|}$$

$$MacroAveragingRecall = \frac{\sum_{c_i \epsilon C} Rcl(D, c_i)}{|c|}$$

We have calculated the ROC curves for each category for every model. These ROC curves for every category was then displayed on a single x-axis and y-axis and micro-average ROC curve was plotted. For, text summarization we will be evaluating the summaries by comparing it with the labels.

7. RESULT

We train 5 different multi-label classifiers for the 9 different categories obtaining the results seen in Table 1.

Table 1: Multi-Label Classification results with Macro and Micro average scores.

Algorithm	ROC_AUC_Macro	ROC_AUC_Micro
BERT-based	53%	56%
CNN	63%	59%
LSTM	70%	71%
Naive Bayes	83%	82%
BiGRU	82%	84%

An example of summarization inference can also be seen below

if a change happens to our business then the new owners may use your personal data in the same way as set out in this privacy notice. we have established a privacy and preferences centre where you can view and make certain decisions about your personal data use. if we need to use your personal data for an unrelated purpose we will notify you and we will explain the legal basis which allows us to do so. internal third parties other companies in the x genomics group acting as joint controllers or processors and who are based in the us.

8. CONCLUSION

As we can see from the results the best performing model is the Bidirectional GRU. The bert based model performs well under its potential. Our hypothesis behind this is that BERT accepts a maximum of 512 tokens so we had to truncate all documents to a max length of 512. In this process we lost a lot of information. Another surprising finding is that simple machine learning algorithm like Naive bayes perform better or as well as the deep learning model. Our hypothesis is that this is due to the smaller size of the dataset and the long lengths of the documents. Unlike sentence classification, the document classification problem has a huge number of words to use to classify each document. This makes the order of the words to have less control over the prediction, causing techniques like CNN and LSTM to be less effective.

9. ROLES OF COLLABORATORS

Topic selection, Researching on topic and project proposal(documentation) everyone has equal contribution in this.
Pranav:- Implemented the Bi-GRU and BERT model for multi label classification and contributed towards document pre-processing and cleanup.
Kesar- Implemented the Multinomial Naive Bayes and LSTM model for multi label classification and contributed towards document pre-processing and cleanup.
Vinay- Implemented the CNN model for for multi label classification and contributed towards topic modelling for summarization.
Siddharth - Contributed towards evaluation of the models and the summarization using topic modelling.

10. REFERENCES

[1] J. E. O. Samir Undavia, Adam Meyers, "A Comparative Study of Classifying Legal Documents with Neural Networks," 2018.

[2] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. LeCun, "Very deep convolutional networks for natural language processing," *CoRR*, vol. abs/1606.01781, 2016.

[4] R. C. Staudemeyer and E. R. Morris, "Understanding lstm – a tutorial into long short-term memory recurrent neural networks," 2019.

[5] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," 2017.