Applying Explainable AI techniques to the APP-350 Corpus - MCS MidTerm Capstone Report

Tristan Koh

October 25, 2022

Abstract

This is the for the capstone project YSS4107 under the double degree law and liberal arts program. It is also intended to fulfil the MCS major capstone requirements. The code can be found here.

1 Definitions

- Explainable AI is a branch of AI that seeks to explain the predictions of machine learning models through statistical explanations and visualisations.
- Natural Language Processing (NLP) is a subset of machine learning that aims to quantifiably analyse natural language.
- APP-350 Corpus is a corpus of 350 Android app privacy policies annotated with privacy practices (i.e. behaviour that can have privacy implications).

2 Objective of capstone

Using Interpret Text, I aim to train a machine learning model using the APP-350 corpus to predict the type of privacy practices within app privacy policies. I then use Interpret Text tools to visualise why the model made such predictions from a machine learning perspective. I then compare and contrast this with how a lawyer would reason about the privacy policy.

3 Rationale of capstone

Natural language forms the bread and butter of the legal industry, as expressed in contracts, judgements and legislation. There has been an increasing trend within the legal industry to adopt more machine learning techniques to automate and assist low level legal analysis. Within the specific context of data privacy, it would be useful

to have a tool that assesses the possible data privacy risks of user policies. Such a tool would naturally use NLP techniques. Nevertheless, these tools should still be accessible to the layperson lawyer that might not be trained in data science.

While NLP techniques have substantially increased in performance in recent years, it has come at the cost of explainability of predictions because of the usage of neural networks which are architecturally more complex than traditional machine learning models. This lack of explainability of neural networks could potentially be a significant hindrance towards their adoption within the legal industry because the lawyer / law firm which uses these models still ultimately bear the responsibility of ensuring that the analysis is legally sound.

However, the intersection in skillset between data science and legal analysis is still nascent and it is unrealistic to expect all legally trained personnel to be trained in data science to the extent required to interpret the predictions of machine learning models without aid.

Recent research within the legal NLP space have focused on building higher performing models and Text representation, but there have been comparatively few papers that assess the explainability of such models. Therefore, my capstone aims to bridge the gap between the lawyer and the data scientist by using Explainable AI techniques to explain the predictions of machine learning models in the context of predicting data privacy practices of app policies.

4 Explanation of Dataset ("The APP-350 corpus")

The APP-350 Corpus consists of 350 annotated Android app privacy policies. Each annotation consists of a practice and a modality.

A "privacy practice" (or "practice") describes a certain behaviour of an app that can have privacy implications (e.g., collection of a phone's device identifier or sharing of its location with ad networks). There are two modalities: PERFORMED (i.e. a practice is explicitly described as being performed) and NOT_PERFORMED (i.e. a practice is explicitly described as not being performed).

As not all practices had modalities, altogether, 57 different categories were annotated. The following is a table of the practices and their descriptions.

Data Type	Description
Contact	The policy describes collection of unspecified contact data.
Contact_Address_Book	The policy describes collection of contact data from a user's address book on the phone.
Contact_City	The policy describes collection of the user's city.
Contact_E_Mail_Address	The policy describes collection of the user's e-mail.
Contact_Password	The policy describes collection of the user's password.
Contact_Phone_Number	The policy describes collection of the user's phone number.
Contact_Postal_Address	The policy describes collection of the user's postal address.
Contact_ZIP	The policy describes collection of the user's ZIP code.
Demographic	The policy describes collection of the user's unspecified demographic data.
Demographic_Age	The policy describes collection of the user's age (including birth date and age range).
Demographic_Gender	The policy describes collection of the user's gender.
Identifier	The policy describes collection of the user's unspecified identifiers.
Identifier_Ad_ID	The policy describes collection of the user's ad ID (such as the Google Ad ID).
Identifier_Cookie_or_similar_Tech	The policy describes collection of the user's HTTP cookies, flash cookies, pixel tags, or similar identifiers.
Identifier_Device_ID	The policy describes collection of the user's device ID (such as the Android ID).
Identifier_IMEI	The policy describes collection of the user's IMEI (International Mobile Equipment Identity).
Identifier_IMSI	The policy describes collection of the user's IMSI (International Mobile Subscriber Identity).
Identifier_IP_Address	The policy describes collection of the user's IP address.
Identifier_MAC	The policy describes collection of the user's MAC address.
Identifier_Mobile_Carrier	The policy describes collection of the user's mobile carrier name or other mobile carrier identifier.
Identifier_SIM_Serial	The policy describes collection of the user's SIM serial number.
Identifier_SSID_BSSID	The policy describes collection of the user's SSID or BSSID.
Location	The policy describes collection of the user's unspecified location data.
Location_Bluetooth	The policy describes collection of the user's Bluetooth location data.
Location_Cell_Tower	The policy describes collection of the user's cell tower location data.
Location_GPS	The policy describes collection of the user's GPS location data.
Location_IP_Address	The policy describes collection of the user's IP location data.
Location_WiFi	The policy describes collection of the user's WiFi location data.
SSO	The policy describes receiving data from an unspecified single sign on service.
Facebook_SSO	The policy describes receiving data from the Facebook single sign on service.

Table 1: List of annotated data privacy practices and their descriptions.

The APP-350 Corpus was used in a broader project to train machine learning models to conduct a privacy census of 1,035,853 Android apps. In that project, the researchers downloaded the data privacy practices of all apps from the Play Store with more than 350 million installs (which totalled 247 apps) and 103 randomly selected apps with 5 million installs. In total, the researchers collected the data privacy policies of 350 apps.

All 350 policies were annotated by one of the authors, a lawyer with experience in data privacy law. To ensure reliability of annotations, 2 other law students were hired to double annotate 10% of the corpus. With a mean of Krippendorff's $\alpha = 0.78$[1], the agreement between the annotations exceeded previous similar research.

For more information about how the Corpus was annotated, see the paper "MAPS: Scaling Privacy Compliance Analysis to a Million Apps", Section 3, Pg 69 to 70.

5 Rationale for utilising the APP-350 corpus

Since the focus of this capstone is to assess the interpretability of XAI models specifically within a legal context, this dataset was chosen for the following reasons:

[1]Krippendorff's α is a measure of agreement, with $\alpha > 0.8$ indicating good agreement, $0.67 <= \alpha <= 0.8$ indicating fair agreement, and $\alpha < 0.67$ indicating doubtful agreement.

1. APP-350 contains real-world data privacy practices as they were scraped from Google PlayStore apps. Thus training XAI models on such a dataset would provide a realistic insight into the extent of which AI models are explainable in the legal context.

2. Legal tech companies are also using such datasets to train models as part of their contract / document review products. By using APP-350 to train XAI models, the results can be used as a (simple)[2] proxy for the explainability of models that are currently used in the industry.

3. APP-350 is a labelled dataset, allowing easy validation of results. If an unlabelled dataset was used, unsupervised training would have to be conducted. The performance of the models would likely be much lower because NLP models for specific vocabulary like law are still not as sophisticated as models trained on general vocabulary. Further, there are few law specific labelled datasets to begin with.

4. APP-350 is labelled on both the sentence and segment (i.e. paragraph) level. This provides more granular data for training the AI models.

6 Methodology

There are three major steps to the capstone: First is data pre-processing, second is model training and applying XAI techniques to visualise their predictions ("Model Training and XAI visualisation"), and the last is to survey law and non-law students about whether they find these explanations interpretable ("Survey to assess interpretability"). I describe the specific methodology of these parts below.

6.1 Data pre-processing

The annotated privacy policies were originally in .yml format, with one .yml file containing one app data privacy policy. As explained above, each data privacy policy is labelled at both the sentence and segment level. The data was restructured from .yml to .csv, with one .csv file containing annotated sentences and the other containing annotated segments. By having two levels of text data for model training, this would provide another dimension to compare model performance on.

6.2 Model training and XAI visualisation

As the original researchers used the same dataset to train classifiers to predict on unseen data privacy policies, I adopt their training methodology and model choice as a guide for this capstone.

[2]The datasets used in industry are usually much larger and the models used are more complicated. However, APP-350 would be sufficiently complicated to serve as a toy example at an undergraduate level.

The original researchers feature engineered the data as follows (Page 71 to 72):

1. Tokenisation: Lowercase all characters, remove non-ASCII characters, no stemming, normalisation of whitespace and punctuation, unigrams and bigrams.
2. Word representation: Union of TF-IDF and manually crafted features. The manually crafted features consist of Boolean values indicating the presence or absence of indicative strings the researchers observed in the data.

Individual classifiers were then trained for every policy classification. For all the classifications (except for four categories), they trained a model using the scikit-learn SVC implementation with a linear kernel, with five-fold cross validation. For the four policy classifications, word-based rule classifiers were used instead because of the limited number of training data.
(Add performance of researchers' models here.)

6.2.1 Proposed model training methodology

There are three main factors that I vary: Text representation, ML model, and XAI package used to explain the trained model.

6.2.2 Text representation

Computers cannot understand text directly and have to be converted into some kind of quantitative data. Therefore in NLP, text representations are methods to represent text as numeric or continuous vectors. This step is done before the model is trained. I use Tf-IDF (term frequency - inverse document frequency) and GloVe word embeddings as the text representations.

The Tf-IDF metric for a word in a document is calculated by multiplying two different metrics:

1. Term frequency (TF) of a word in a document. This is the number of times the word appears in a document.
2. Inverse document frequency (IDF) of the word across a set of documents. This is calculated by taking the total number of documents and dividing it by the number of documents that contain the specific word. This calculates the rarity of the term across all the documents. The closer the IDF of a word is to 0, the more common the word is.

Mathematically, the Tf-IDF score for the word t in the document d from the document set D can be stated as such:

$$tf - idf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

where

$$tf(t,d) = log(1 + freq(t,d))$$
$$idf(t,D) = log\left(\frac{N}{count(d \in D : t \in d)}\right)$$

While Tf-IDF is easy to calculate, one of its limitations is that it is a purely count-based metric. Tf-IDF does not take into account the context of the word. For example, Tf-IDF would not be able to capture the semantic relationship between words. Word embeddings try to overcome this issue with count-based metrics like Tf-IDF. Word embeddings are vector representation of words, such that the vectors closer to each other in a vector space are similar in their semantic meaning.[3]

GloVe is one type of word embeddings (more information to be added)

6.2.3 Model choice

As the researchers found that the SVC classifier produces the best performance, I use SVC as well. In all, I use the following models:

1. Logistic regression
2. SGDClassifier
3. SVC
4. Ensemble classifiers (AdaBoost, GradientBoost, Random Forest)

Logistic regression functions as a baseline classifier for its simplicity. SGDClassifier functions a possible alternative to SVC since the SGDClassifier can use a linear SVM loss function. Ensemble classifiers are included to provide a wider range of models to compare performance.

The two broad types of AI models that are usually used are classical machine learning models (such as logistic regression and tree-based classifiers) and neural networks. Though recent advances in NLP are in the field of neural networks [4], I chose to focus only on classical ML models to reduce the possible complexity of the capstone, as the field of NLP by itself produces models that are usually more complex than models trained on quantitative variables. Explaining neural networks used for NLP would be a much more complex task compared to explaining classical ML models.
Further, as the APP-350 corpus only contains (insert number here) datapoints, there is insufficient data to train neural networks. Generally, neural networks require (add number here) datapoints to perform well.

[3] https://becominghuman.ai/mathematical-introduction-to-glove-word-embedding-60f24154e54c
[4] State of the art NLP models include BERT and ELMo.

6.2.4 XAI packages

1. LIME
2. SHAP
3. OmniXAI
4. Interpret Text

   Interpret Text allows visualisation of the predictions of NLP models across a range of complexities.

   For linear and tree-based models, Interpret Text uses a Classical Text Explainer. The Explainer leverages this inherent explainability by exposing weights and importances over encoded tokens as explanations over each word in a document. In practice, these can be accessed through the visualization dashboard or the explanation object.

   The explanations provided by the aforementioned glass-box methods serve as direct proxies for weights and parameters in the model, which make the final prediction.

6.3 Survey to assess interpretability

To be added.

7 Findings so far

7.1 Exploratory Data Analysis (EDA)

As mentioned above, there are two levels of text data: Sentence level and segment level. As I train and compare the performance of models on both levels of data, I also conducted the EDA on both levels.

7.1.1 Sentence level

There are a total of 18829 annotated sentences. The table below shows some summary statistics of the top 10 and bottom 10 frequently occurring practices at the sentence level.

practice	counts	sentence_length_mean	sentence_length_median	counts_percentage
Identifier_Cookie_or_similar_Tech_1stParty	2107	25.389654	22.0	11.2%
Contact_E_Mail_Address_1stParty	2106	28.651472	25.0	11.2%
Location_1stParty	1514	29.159181	24.0	8.1%
Identifier_Cookie_or_similar_Tech_3rdParty	1250	27.318400	24.0	6.6%
Identifier_IP_Address_1stParty	1005	30.913433	27.0	5.3%
Contact_Phone_Number_1stParty	970	29.117526	25.0	5.2%
Identifier_Device_ID_1stParty	697	32.377331	28.0	3.7%
Contact_Postal_Address_1stParty	597	28.907873	26.0	3.2%
SSO	504	32.565476	28.0	2.7%
Demographic_Age_1stParty	428	33.074766	26.0	2.3%

Table 2: Summary statistics for top 10 occurring practices at sentence level.

practice	counts	sentence_length_mean	sentence_length_median	counts_percentage
Identifier_Mobile_Carrier_3rdParty	35	47.057143	30.0	0.19%
Contact_ZIP_3rdParty	34	40.176471	41.0	0.18%
Identifier_SSID_BSSID_1stParty	33	28.060606	24.0	0.18%
Contact_Password_3rdParty	33	24.181818	20.0	0.18%
Contact_City_3rdParty	24	18.000000	14.0	0.13%
Contact_Address_Book_3rdParty	17	39.647059	34.0	0.1%
Identifier_IMSI_1stParty	13	48.153846	44.0	0.07%
Identifier_SIM_Serial_3rdParty	5	41.200000	54.0	0.03%
Identifier_IMSI_3rdParty	4	54.250000	47.5	0.02%
Identifier_SSID_BSSID_3rdParty	2	65.500000	65.5	0.01%

Table 3: Summary statistics for bottom 10 frequently occurring practices.

In total, the top 10 frequently occurring practices make up approximately 60% of the dataset. The bottom 10 frequently occurring practices make up approximately 1% of the dataset.

According to Table 4 below, there does not seem to be much variation in sentence length for the top 10 frequently occurring practices, since the standard deviation for the mean is approximately 2.5 words and the median 1.9 words. This could indicate similar sentence complexity across the practices.

	sentence_length_mean	sentence_length_median
count	10.000000	10.000000
mean	29.747511	25.500000
std	2.468191	1.900292
min	25.389654	22.000000
25%	28.715572	24.250000
50%	29.138353	25.500000
75%	32.011357	26.750000
max	33.074766	28.000000

Table 4: Summary sentence statistics for top 10 frequently occurring practices.

7.1.2 Segment level

There are in total 21623 segments. However, there are 11422 segments without any annotated practice. Hence there are 10201 annotated segments. The table below shows some summary statistics of the top 10 and bottom 10 frequently occurring practices at the segment level.

practice	counts	segment_length_mean	segment_length_median	counts_percentage
Contact_E_Mail_Address_1stParty	1105	84.118552	72.0	10.83
Identifier_Cookie_or_similar_Tech_1stParty	858	81.913753	68.5	8.41
Location_1stParty	821	89.794153	70.0	8.05
Identifier_IP_Address_1stParty	590	96.984746	73.0	5.78
Contact_Phone_Number_1stParty	565	90.371681	67.0	5.54
Identifier_Cookie_or_similar_Tech_3rdParty	524	100.339695	86.5	5.14
Identifier_Device_ID_1stParty	446	96.704036	72.0	4.37
Contact_Postal_Address_1stParty	364	85.598901	69.5	3.57
SSO	274	99.463504	86.5	2.69
Demographic_Age_1stParty	259	96.200772	80.0	2.54

Table 5: Summary statistics of top 10 frequently occurring practices by segment.

practice	counts	segment_length_mean	segment_length_median	counts_percentage
Identifier_IMEI_3rdParty	23	115.347826	93.0	0.23
Identifier_Mobile_Carrier_3rdParty	21	142.000000	130.0	0.21
Contact_Password_3rdParty	18	70.555556	65.0	0.18
Identifier_SSID_BSSID_1stParty	16	86.062500	71.0	0.16
Contact_Address_Book_3rdParty	14	296.928571	78.5	0.14
Identifier_IMSI_1stParty	11	92.363636	78.0	0.11
Contact_City_3rdParty	8	100.500000	104.0	0.08
Identifier_SIM_Serial_3rdParty	3	85.333333	65.0	0.03
Identifier_IMSI_3rdParty	3	62.333333	65.0	0.03
Identifier_SSID_BSSID_3rdParty	2	105.500000	105.5	0.02

Table 6: Summary statistics for bottom 10 frequently occurring practices.

The top 10 practices make up approximately 57% of the dataset. The bottom 10 practices make up approximately 1.2% of the dataset.

According to Table 7 below, there does not seem to be much variation in sentence length for the top 10 frequently occurring practices, since the standard deviation for the mean is approximately 6.7 words and the median 7.2 words. This could indicate similar segment complexity across the practices.

	segment_length_mean	segment_length_median
count	10.000000	10.000000
mean	92.148979	74.500000
std	6.683275	7.230337
min	81.913753	67.000000
25%	86.647714	69.625000
50%	93.286227	72.000000
75%	96.914568	78.250000
max	100.339695	86.500000

Table 7: Summary segment statistics for top 10 frequently occurring practices.

7.2 Performance of classifiers for top N practices

As seen in Table 2 and Table 5 above, there is an uneven distribution of records across the practices. Further, the bottom 10 frequently occurring practices for both sentence and segment level only contains about 2 to 35 records for each practice. Given that there are in total 57 practices for the entire dataset, and there is not a uniform distribution of occurrences. Training models on all 57 practices would likely lead to low performance since there are not enough records for all 57 practices. Thus, to find an optimal balance between model performance and still maintain a realistic sample of practices that could appear in a real world dataset, I chose to assess model performance by first assessing the performance of the models for the top N (where $3 <= N <= 10$) frequently occurring practices at both the sentence and segment level.

I use Logistic Regression, SGDClassifier and SVC classifiers and compare the weighted precision, recall and F1 scores. Precision, Recall and F1 scores different metrics are used to assess the performance of classifiers. They are stated mathematically below.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision + Recall}}$$

Generally, precision is the preferred metric when the cost of false positives are high, such as detecting spam. If a classifier classifies a non-spam email as spam, the user would lose important information. Whereas recall is preferred when the costs of false negatives are high. For example, if a classifier classifies someone as not having cancer when they actually have cancer, the patient would lose the opportunity for early intervention. F1 is a harmonic mean of precision and recall, and is usually

used to find a balance between precision and recall. Since there is neither a high cost for false positives or false negatives, F1 score is primarily used to assess the performance across the top N practices. As the distributions of records across the practices are uneven, I focus on the weighted average of F1 scores.
I also assess the top N performance for both the Tf-IDF and GLoVe word embeddings.

7.2.1 Sentence level performance - Tf-IDF

Generally we see that the SVC classifier performance the best across the metrics and across the top N frequently occurring practices. This corresponds with the findings by the researchers as they also found that the SVC classifier produced the best performance.

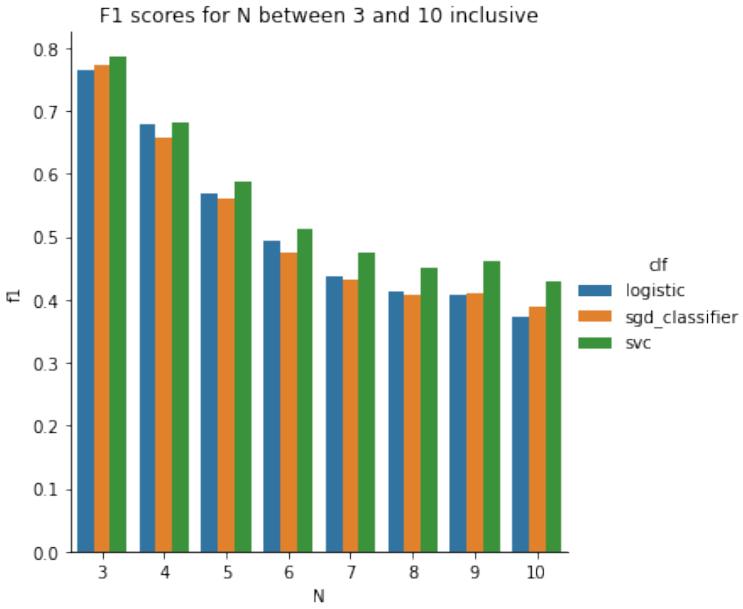

Figure 1: F1 scores for classifiers for top N occurring practices at sentence level

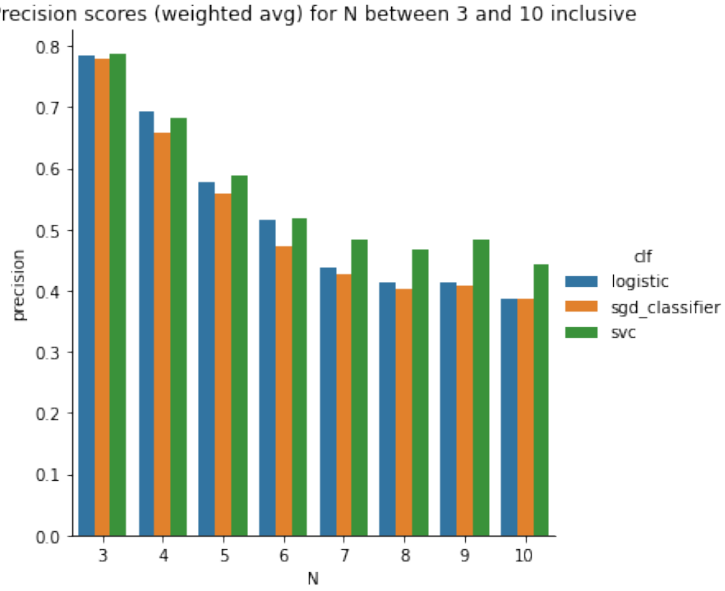

Figure 2: Precision scores for classifiers for top N occurring practices at sentence level

7.2.2 Sentence level performance - Word embeddings

7.2.3 Segment level performance - Tf-IDF

7.2.4 Segment level performance - Word embeddings

7.3 Performance of individual classifiers for top 5 practices

7.4 XAI visualisation of models

8 Further steps to take in Semester 2

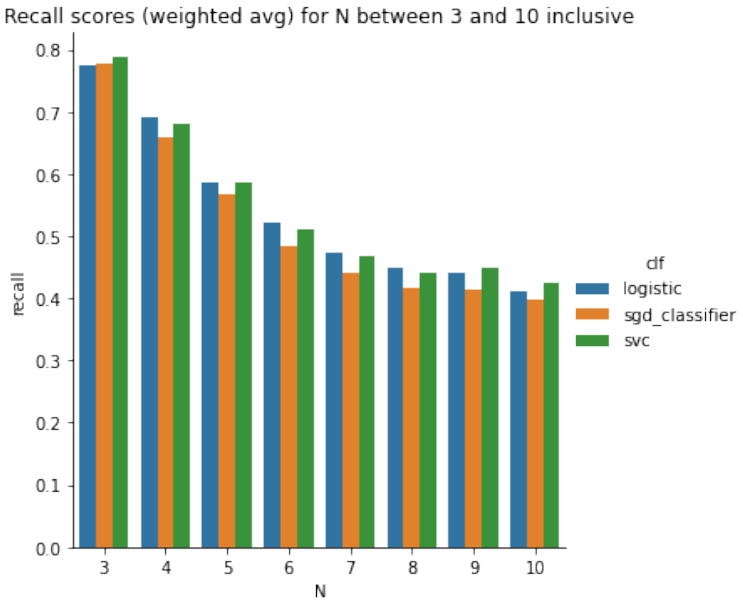

Figure 3: Recall scores for classifiers for top N occurring practices at sentence level