Applying Explainable AI techniques using Interpret Text to the APP-350 Corpus

Tristan Koh

August 2, 2022

Abstract

This is a research proposal 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.

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.
- Interpret Text is a Python framework developed by Microsoft that makes it easier to interpret the predictions of NLP models.
- 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 artificial neural networks which are architecturally more complex than traditional machine learning models. This lack of explainability of more advanced machine learning models 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 also unrealistic all legally trained personnel to be trained in data science to the extent required to interpret the predictions of machine learning models without aid.

Therefore, my capstone aims to bridge the gap between the lawyer and the data scientist by using Interpret Text to explain the predictions of a machine learning model in the context of assessing data privacy practices of app policies.

4 More about APP-350

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). Altogether, 58 different practices were annotated.

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).

Below is a screenshot 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.

Party	Description
1stParty	The policy describes collection of data from the user by the app publisher.
3rdParty	The policy describes collection of data from the user by ad networks, analytics services, or other third parties.

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

5 Rationale for utilising the APP-350 corpus

Since the focus of this capstone is to assess the manner in which legal analytics can transform legal practice and its corresponding limitations, this dataset was chosen for the following reasons:

1. 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.

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

3. APP-350 contains real-world data privacy practices. 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.

4. Legal tech companies are also using classification models to assist in contract / document review products. To do so, these companies train their AI models in a similar manner on similarly structured datasets. By using APP-350 to train XAI models, the results can be used as a proxy for the explainability of models that are currently used in the industry.

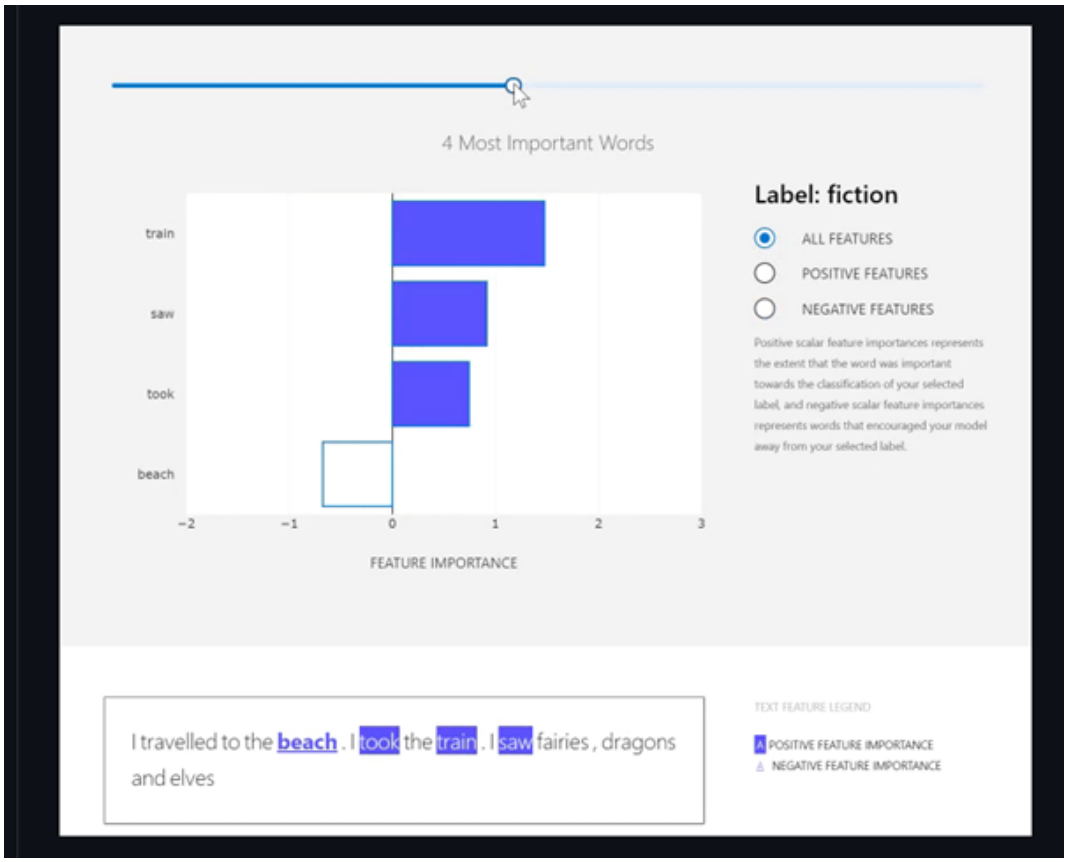
6 How Interpret Text explains predictions

Interpret Text allows visualisation of the predictions of NLP models across a range of complexities. The two broad types of models that it covers are linear and tree-based models and neural networks. For the purposes of this capstone, I will mainly be focusing on the first type of model as these models are considered "glass-box" models: A glass-box explainer implies a model that is innately explainable, where the user can fully observe and dissect the process adopted by the model in making a prediction. The hyperparameters of neural networks are much more complicated which may be too technically complicated for this capstone.

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.

This is how the visualisation dashboard looks like:



The user can visualise the top 10 important words (both positive and negative) for each label, and the dashboard provides a "weight" score for each word. The user can interact with the highlighted words to see the individual score.

7 Model training methodology by researchers

According to the paper "MAPS: Scaling Privacy Compliance Analysis to a Million Apps", the researchers feature engineered the policy text 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.
   (a) 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 policy classifications), they trained a model using

scikitlearn 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.

8 Proposed model training methodology

As the methodology used by the researchers seem to produce the highest performing models, I aim to replicate it. However, since Interpret Text only supports linear and tree-based classifiers, I will use the SGDClassifier instead to fit a linear SVM. I then will use Interpret Text to visualise the predictions.

Apart from replicating the researchers' methodology, I will also experiment with different feature engineering approaches to compare their impact on model performance and explainability (further details to be added):

1. Word representations: Bag of words and word embeddings.
2. Different combinations of n-grams.

I will also assess the performance and explainability of different scikitlearn linear and tree-based models (further details to be added):

1. Logistic regression
2. Ridge classifier
3. Ada Boost
4. Gradient Boost
5. Random forest