YaleNUSCollege

TRANSPARENCY IN THE MACHINE: APPLYING & EVALUATING EXPLAINABLE AI TECHNIQUES IN LEGAL DECISION MAKING

TRISTAN KOH LY WEY

Capstone Final Report for Yale-NUS BSc (Honours) in Mathematical, Computational and Statistical Sciences & LLB (Honours) Double Degree Program

Supervised by:

Professor Michael Choi and Professor Simon Chesterman

AY 2022/2023

Yale-NUS College Capstone Project

DECLARATION & CONSENT

1. I declare that the product of this Project, the Thesis, is the end result of my own work and that due acknowledgement has been given in the bibliography and references to ALL sources be they printed, electronic, or personal, in accordance with the academic regulations of Yale-NUS College.

2. I acknowledge that the Thesis is subject to the policies relating to Yale-NUS College Intellectual Property (Yale-NUS HR 039).

ACCESS LEVEL

3. I agree, in consultation with my supervisor(s), that the Thesis be given the access level specified below: [check one only]

   o Unrestricted access
   Make the Thesis immediately available for worldwide access.

   o Access restricted to Yale-NUS College for a limited period
   Make the Thesis immediately available for Yale-NUS College access only from _____ (mm/yyyy) to _____ (mm/yyyy), up to a maximum of 2 years for the following reason(s): (please specify; attach a separate sheet if necessary):
   _____.

   After this period, the Thesis will be made available for worldwide access.

   o Other restrictions: (please specify if any part of your thesis should be restricted)
   _____
   _____

_____
Name & Residential College of Student

_____ _____
Signature of Student Date

_____ _____
Name & Signature of Supervisor Date

Acknowledgements

This capstone would not be possible if not for the gracious provision of the following individuals and groups, and most of all, God.

YALE-NUS COLLEGE

Abstract

B.Sc (Hons) and L.L.B (Hons)

Transparency in the Machine: Applying & Evaluating Explainable AI in Legal Decision Making

by Tristan KOH

This capstone assesses the explainability of the visualisations of machine learning models that were trained to identify data practices within apps' data privacy policies.

In recent years, the performance of machine learning models have increased but have come at the cost of decreased explainability. Machine learning has been gradually adopted in the legal industry to assist with low level legal analysis. This lack of explainability could be a significant hindrance towards greater adoption of artificial intelligence (AI) because the lawyer and law firm that use these models ultimately bear the responsibility of ensuring that their advice is legally accurate.

Separately, data privacy has been subjected to increased regulative oversight. With more ways to collect and use personal data, organisations are more likely to misuse such data. Therefore, users' awareness about how their data is collected and used should be heightened. Similarly, regulators and organisations have increased obligations to ensure that data is being collected and used responsibly.

This capstone uses this data privacy context to train reasonably performing models that are able to detect data practices within apps' data privacy practices. As the predictions of these models could be helpful to users, regulators and organisations, Explainable AI (XAI) techniques are applied on these models to produce visualisations that aim to explain these predictions to these stakeholders that may not be experts in data science. Finally, the effectiveness of these visualisations are assessed by surveying Yale-NUS and NUS students about their self-reported levels of trust, fairness and perceptions of effectiveness of these visualisations.

"In the doorway of my Father's house I'm Home."

Psalm 84 (I'm Home) by Shane and Shane

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and significance

Natural language forms the bread and butter of the legal industry, as it is expressed in contracts, judgements and legislation. The legal industry has been adopting more machine learning tools to automate and assist low level legal analysis. Worldwide legal tech market revenues were at 27.6 billion USD and is projected to to grow at a compound annual growth rate of 4% to 35.6 billion USD by 2027 (Statista, 2022). As early as 2018, LawGeex, a contract review startup, compared the performance of lawyers vs LawGeex's machine learning model in reviewing standard template Non-Disclosure Agreements (NDA). The model beat the humans both in terms of accuracy and time, with the model having a 94% accuracy rate and taking 26 seconds to complete the review. In comparison, the lawyers had an average accuracy of 85% and took 92 minutes to finish the task (LawGeex, 2018). More significantly, at the end of 2022, an AI model GPT [1] took the US bar examination and got 50% of

[1]The GPT which was trained by OpenAI forms the foundation of the now famous ChatGPT, which will be mentioned below.

the questions correct, and performed at a passing rate for both Evidence and Torts (Bommarito and Katz, 2022).

Like the models that were trained by LawGeex and OpenAI, most legal tech tools that aim to conduct low level legal analysis use natural language processing (NLP) techniques. NLP is a branch of AI that is concerned with giving computers the ability to understand text and spoken words in much the same way human beings can (IBM, 2022). While NLP techniques have substantially increased in performance in recent years, it has come at the cost of explainability of their predictions because of models that are architecturally more complex (El Zini and Awad, 2022). This lack of explainability 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 legal responsibility of ensuring that the analysis is legally sound.

Nevertheless, 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. Research within the legal NLP space have focused on building higher performing models, but there have been comparatively few papers that assess the explainability of such models. At the same time, explainable AI (XAI) techniques and research have been rising in popularity since 2020 (Linardatos, Papastefanopoulos, and Kotsiantis, 2020) but have not been specifically applied onto legal text. Therefore, this project 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.

Separately, the widespread collection and use of data by organisations in recent years has led to an increase of regulations governing data privacy. This "datafication" of society includes the "transformation of interactions into data that can be valued and used for predictive analysis". Governments have therefore stepped up their efforts to guarantee privacy, with 145 countries having enacted data protection legislation in 2021 (Gstrein and Beaulieu, 2022). With more sophisticated regulation comes increased difficulties for organisations to ensure that they are complying with these regulations, and for governments to enforce them. A possible area of legal tech would be tools to aid in the compliance of these regulations. Therefore, I focus on NLP and XAI in the specific context of data privacy. This context provides a realistic evaluation of the interpretability of models that are trained on legal texts relating to data privacy.

1.1.1 Development and importance of explainable AI in legal technology

With the recent release of ChatGPT (OpenAI, 2022) to the general public at the start of 2023,

1.1.2 Development and importance of data privacy regulation

1.2 Explanation of the APP-350 Corpus

The APP-350 Corpus consists of 350 annotated Android app privacy policies. The corpus has been used by a previous paper to train models to detect data privacy practices (Zimmeck et al., 2019). 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.

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

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 1.1: List of annotated data privacy practices and their descriptions.

a mean of Krippendorff's $\alpha = 0.78$[2], 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.

1.2.1 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. APP-350 contains real-world data privacy practices as they were scraped from Google PlayStore apps. Thus training XAI models on

[2]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.

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)[3] 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.

[3]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.

1.3 Problem statement

1.4 Main findings and roadmap

1.5 Font Formatting Commands

Similarly to Word, LaTeX provides simple formatting, including **bold**, *italic*, underlined and `ugly stuff`. However, no underline or strikethrough by default. You can also change the size of the text, using tiny, small, large, huge. These last commands work within a specific scope. The scope can be specified using { and }, with the { placed before the \size command.

1.5.1 Special characters

In that case, simply use $ (by the way, note that using the dollar sign in your text switches to mathematical notation. To actually print a dollar sign use the \textdollar command). The equation above has a label, meaning you can refer to it. The numbering system uses the chapter number (in this case 1), then the equation position within the chapter (1 again). Example: Equation **??** is an example of an equation in LaTeX. In case you would like to have an equation without numbering it? Easy!

$$t = a \times log_2(\frac{D}{W} + 1) + b$$

The only difference? The $*$ symbol in the \begin{equation*}. This also works with Figures and Tables.

1.6 Code Snippets

```
int main (int argc, char ** argc)
{
  printf("Hello world!\n");
  return 0;
}
```

This template uses the `lstlisting` package, which not the best for code snippets. However, it works without any problem, while other packages may have compatibility issues. Feel free to try alternative solutions, the best one being `minted`.

1.7 Figures

Figures are a bit tricky with LaTeX (not as much as tables though). Let us see a simple example below: You can refer to it: Figure 1.1. This is possible thanks to the \label command. The figure should also be shown on the List of Figures page (note this other way of referring to another part of the manuscript!). A common practice is use the following naming convention:

- A prefix, indicating the nature of the object labelled: `eq` for equations, `fig` for figures, `tab` for tables.
- A colon.
- A unique name (easy to remember) describing your figure. Example: exp1confmatrix would suggest that the figure shows a confusion matrix for your experiment 1.

FIGURE 1.1: When a YNC alumni tells you that back in their days, they did not have LaTeX template and would write their report in latin on a papyrus.

A few other points: The `\caption` and `\label` can be put either before or after the `\includegraphics` command. When you create a Figure, you need to provide placement information for LaTeX. LaTeX will usually not locate the figures *exactly* where you want them. The most common specifiers are: `h` (here), `b` (bottom of the page) and `t` (top). The `!` specifier tries to force LaTeX to put the image exactly at the location you specified (with mixed success though). For a longer list of specifiers, please refer to: https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.

1.7.1 Figure Size

The size of the figure can be determined by the first parameter of the `\includegraphics` command. In this example, we set the size to be $0.9 \times$

`textwidth`, or 90% of the size of a column. We could have used an absolute value in cm, e.g. `width=19cm`.

1.7.2 Supported Formats

Use standard formats, such as PNG, PDF, JPG. LaTeX also supports other formats, such as EPS. **Rule of thumb: use PDF as much as you can, as it uses vector graphics, making it easy to scale the figure to very large format without problems.**

1.7.3 Multiple images in one figure

You can also create complex figures with multiple images. Here is an example, which uses a 2×2 layout. The overall figure can be referred as Figure 1.2.

1.8 Tables

Tables can be a nightmare in LaTeX. The easiest way to deal with tables in LaTeX is to use some online tools. My favorite so far: `https://www.tablesgenerator.com/`

Anyway, for Tables, using the LaTeX Table Generator is a great option.

1.8.1 How to get Bibtex References?

The easiest way to find the Bibtex snippet you need for a given reference is to use Google Scholar (**Scholar**). On the main page, type the name of

FIGURE 1.2: Example of a complex figures on a 2×2 layout.

the paper you are looking for.

In the results page, locate the paper:

FIGURE 1.3: Example of result on Scholar

On the last line of the result (shown in Figure 1.3), there is a " symbol. Clicking on it will display a pop-up.

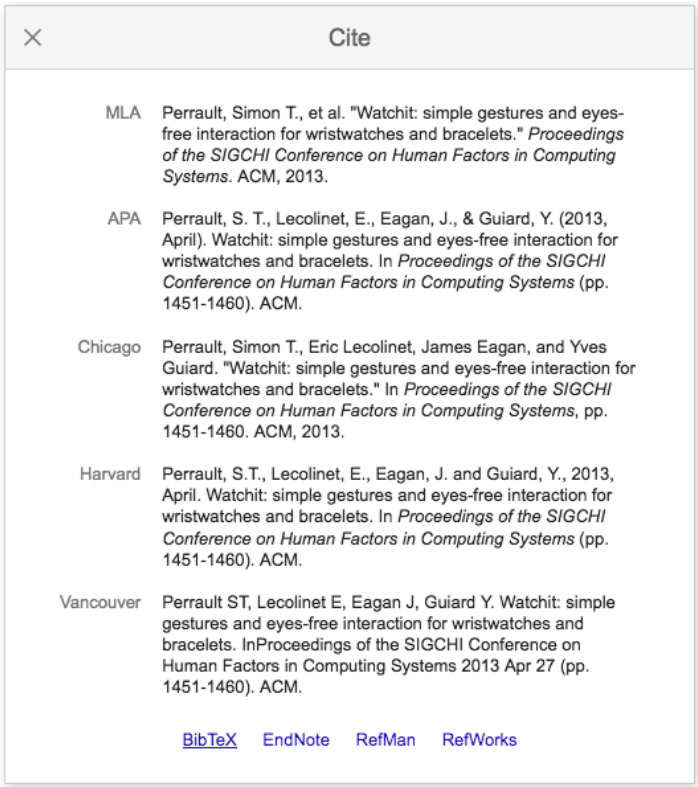

FIGURE 1.4: Pop-up window with the possible citations

At the bottom (see Figure 1.4), you will notice a "Bibtex" link. Click on it. Scholar will then display a small block of text starting with @ symbol. Copy and paste this snippet in your `biblio.bib` and you are done. You may eventually want to check the citation key to something shorter.

You may get unexpected compilation errors with some references. The most common case is that the bibtex entry contains a DOI field, which in turn contains an underscore (_). If that is the case, simply remove the DOI field (not a great practice but a good workaround).

Chapter 2

Review of current literature

We can reference other chapters, for example, here we refer to Chapter 1.

2.1 Training models for NLP

2.2 XAI methods for NLP

Welcome to this LaTeX Thesis Template, a beautiful and easy to use template for writing a thesis using the LaTeX typesetting system.

2.3 Evaluating the effectiveness of XAI methods

2.3.1 User validation of explanations

2.4 Thesis Features and Conventions

To get the best out of this template, there are a few conventions that you may want to follow.

One of the most important (and most difficult) things to keep track of in such a long document as a thesis is consistency. Using certain conventions and ways of doing things (such as using a Todo list) makes the job easier. Of course, all of these are optional and you can adopt your own method.

2.4.1 References

The `biblatex` package is used to format the bibliography and inserts references such as this one (**Reference1**). The options used in the `main.tex` file mean that the in-text citations of references are formatted with the author(s) listed with the date of the publication. Multiple references are separated by semicolons (e.g. (**Reference2**; **Reference1**)) and references with more than three authors only show the first author with *et al.* indicating there are more authors (e.g. (**Reference3**)). This is done automatically for you. To see how you use references, have a look at the `Chapter1.tex` source file. Many reference managers allow you to simply drag the reference into the document as you type.

The bibliography is typeset with references listed in alphabetical order by the first author's last name. This is similar to the APA referencing style. To see how LaTeX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links in in-text citations).

2.4.2 Tables

Tables are an important way of displaying your results, below is an example table which was generated with this code:

TABLE 2.1: The effects of treatments X and Y on the four groups studied.

Groups	Treatment X	Treatment Y
1	0.2	0.8
2	0.17	0.7
3	0.24	0.75
4	0.68	0.3

```
\begin{table}
\caption{The effects of treatments X and Y on the four groups studied.}
\label{tab:treatments}
\centering
\begin{tabular}{l l l}
\toprule
\tabhead{Groups} & \tabhead{Treatment X} & \tabhead{Treatment Y} \\
\midrule
1 & 0.2 & 0.8\\
2 & 0.17 & 0.7\\
3 & 0.24 & 0.75\\
4 & 0.68 & 0.3\\
\bottomrule\\
\end{tabular}
\end{table}
```

You can reference tables with `\ref{<label>}` where the label is defined within the table environment. See `Chapter1.tex` for an example of the label and citation (e.g. Table 2.1).

2.4.3 Figures

There will hopefully be many figures in your thesis (that should be placed in the *Figures* folder). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}
\centering
\includegraphics{Figures/Electron}
\decoRule
\caption[An Electron]{An electron (artist's impression).}
\label{fig:Electron}
\end{figure}
```

Also look in the source file. Putting this code into the source file produces the picture of the electron that you can see in the figure below.

Sometimes figures don't always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so LaTeX puts it at the top of the next page. Positioning figures is the job of LaTeX and so you should only worry about making them look good!

Figures usually should have captions just in case you need to refer to them (such as in Figure 2.1). The `\caption` command contains two parts, the first part, inside the square brackets is the title that will appear in the *List of Figures*, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

FIGURE 2.1: An electron (artist's impression).

The `\decoRule` command is optional and simply puts an aesthetic horizontal line below the image. If you do this for one image, do it for all of them.

LaTeX is capable of using images in pdf, jpg and png format.

2.5 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. LaTeX automatically builds a table of Contents by looking at all the `\chapter{}`, `\section{}` and `\subsection{}` commands you write in the source.

The Table of Contents should only list the sections to three (3) levels. A `chapter{}` is level zero (0). A `\section{}` is level one (1) and so a `\subsection{}` is level two (2). In your thesis it is likely that you will even use a `subsubsection{}`, which is level three (3). The depth to which the Table of Contents is formatted is set within `MastersDoctoralThesis.cls`. If you need this changed, you can do it in `main.tex`.

Chapter 3

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.

3.1 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.

3.2 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.

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.)

3.2.1 Proposed model training methodology

There are three main components to the model training methodology: Text representation, ML model, and XAI package used to explain the trained model.

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.[1]

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

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. SVC
3. 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.

[1] https://becominghuman.ai/mathematical-introduction-to-glove-word-embedding-60f24154e54c

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 [2], 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.

XAI method and package

Local Interpretable Model-agnostic Explanations (LIME) is used in this capstone because it can be fitted to any model and is one of the few XAI packages that is easily implementable.

(some description of how LIME works here)

[2] State of the art NLP models include BERT and ELMo.

Chapter 4

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.

4.1 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.

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.

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 4.1: 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 4.2: Summary statistics for bottom 10 frequently occurring 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 4.3: Summary sentence statistics for top 10 frequently occurring practices.

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.

4.2 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 4.4: 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 4.5: 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 4.6: Summary segment statistics for top 10 frequently occurring practices.

Chapter 5

Performance of models

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.

5.1 Models used and classification metrics

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.

5.1.1 Sentence level performance - TfIDF

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.

(add figures here)

5.1.2 Sentence level performance - GloVe embeddings

5.1.3 Segment level performance - TfIDF

5.1.4 Segment level performance - GloVe embeddings

5.2 Performance of individual classifiers for top 5 practices

Bibliography

Bommarito, Michael James and Daniel Martin Katz (2022). "GPT Takes the Bar Exam". In: Available at SSRN. DOI: https://dx.doi.org/10.2139/ssrn.4314839.

El Zini, Julia and Mariette Awad (2022). "On the Explainability of Natural Language Processing Deep Models". In: *ACM Computing Surveys* 55.103, pp. 1–31. DOI: https://doi.org/10.1145/3529755.

Gstrein, Oskar J. and Anne Beaulieu (2022). "How to protect privacy in a datafied society? A presentation of multiple legal and conceptual approaches". In: *Philos Technol* 35.1, p. 3. DOI: https://doi.org/10.1007%2Fs13347-022-00497-4.

IBM (2022). *What is natural language processing?* Last Accessed: 2023-02-11. URL: https://www.ibm.com/sg-en/topics/natural-language-processing.

LawGeex (2018). *Comparing the Performance of Artificial Intelligence to Human Lawyers in the Review of Standard Business Contracts*. Law Accesssed: 2023-02-11. URL: https://images.law.com/contrib/content/uploads/documents/397/5408/lawgeex.pdf.

Linardatos, Pantelis, Vasilis Papastefanopoulos, and Sotiris Kotsiantis (2020). "Explainable AI: A Review of Machine Learning Interpretability Methods". In: *Entropy (Basel)* 23.1, p. 18. DOI: https://dx.doi.org/10.3390/e23010018.

OpenAI (2022). *ChatGPT: Optimizing Language Models for Dialogue*. Last Accessed: 2023-02-11. URL: https://openai.com/blog/chatgpt/.

Statista (2022). *Legal tech market revenue worldwide from 2021 to 2027*. Last Accessed: 2023-02-11. URL: https://www.statista.com/statistics/1155852/legal-tech-market-revenue-worldwide/.

Zimmeck, Sebastian et al. (2019). "MAPS: Scaling Privacy Compliance Analysis to a Million Apps". In: *Proceedings on Privacy Enhancing Technologies* 2019.3, pp. 66–86.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.