



**APPLYING EXPLAINABLE AI
TECHNIQUES ON THE APP-350
CORPUS**

TRISTAN KOH LY WEY

**Capstone Final Report for BSc (Honours) in
Mathematical, Computational and Statistical Sciences**

**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]

☐ Unrestricted access

Make the Thesis immediately available for worldwide access.

☐ 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.

☐ 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)

Applying Explainable AI Techniques on the APP-350 Corpus

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

Acknowledgements	ii
Abstract	iv
1 Introduction	1
1.1 Motivation and significance	1
1.1.1 Development and importance of explainable AI in legal technology	2
1.1.2 Development and importance of data privacy reg- ulation	3
1.2 Explanation of the APP-350 Corpus	3
1.2.1 Rationale for utilising the APP-350 corpus	5
1.3 Problem statement	6
1.4 Main findings and roadmap	6
1.5 Font Formatting Commands	6
1.5.1 Special characters	6
1.6 Code Snippets	8
1.7 Figures	9
1.7.1 Figure Size	10
1.7.2 Supported Formats	10
1.7.3 Multiple images in one figure	10

1.8	Tables	11
1.9	Bibliography	12
1.9.1	How to get Bibtex References?	13
1.10	End of the Tips	14
2	Review of current literature	15
2.1	Training models for NLP	15
2.2	XAI methods for NLP	15
2.3	Evaluating the effectiveness of XAI methods	16
2.3.1	User validation of explanations	16
2.4	What this Template Includes	17
2.4.1	Folders	17
2.4.2	Files	18
2.5	Thesis Features and Conventions	20
2.5.1	References	21
	A Note on bibtex	21
2.5.2	Tables	22
2.5.3	Figures	23
2.6	Sectioning and Subsectioning	25
2.7	In Closing	25
3	Methodology	26
3.1	Data pre-processing	26
3.2	Model training and XAI visualisation	27
3.2.1	Proposed model training methodology	27
	Text representation	28
	Model choice	29

XAI method and package	30
4 Exploratory Data Analysis (EDA)	31
4.1 Sentence level	31
4.2 Segment level	32
5 Performance of models	35
5.1 Models used and classification metrics	35
5.1.1 Sentence level performance - TfIDF	37
5.1.2 Sentence level performance - GloVe embeddings . .	37
5.1.3 Segment level performance - TfIDF	37
5.1.4 Segment level performance - GloVe embeddings . .	37
5.2 Performance of individual classifiers for top 5 practices . .	37
Bibliography	38
A Frequently Asked Questions	39
A.1 How do I change the colors of links?	39

List of Tables

1.1	List of annotated data privacy practices and their descriptions.	4
1.2	Post-hoc comparisons between body parts. - shows no significant difference ($p > .05$), * shows differences ($p < .05$).	12
2.1	The effects of treatments X and Y on the four groups studied.	23
4.1	Summary statistics for top 10 occurring practices at sentence level.	31
4.2	Summary statistics for bottom 10 frequently occurring practices.	32
4.3	Summary sentence statistics for top 10 frequently occurring practices.	32
4.4	Summary statistics of top 10 frequently occurring practices by segment.	33
4.5	Summary statistics for bottom 10 frequently occurring practices.	33
4.6	Summary segment statistics for top 10 frequently occurring practices.	34

List of Figures

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.	9
1.2	Example of a complex figures on a 2×2 layout.	11
1.3	Example of result on Scholar	13
1.4	Pop-up window with the possible citations	14
2.1	An Electron	24

Chapter 1

Introduction

1.1 Motivation and significance

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.

1.1.1 Development and importance of explainable AI in legal technology

In this chapter, I will demonstrate a few interesting features of LaTeX, and more importantly, provide examples of Figures, Tables, Equations and Code Snippets. These may be easy to deal with on Word, but here it is another story. The general idea is to keep these examples, copy-paste them, then modify them to fit your needs. If you are seeing this content while reading `main.pdf`, please load the overall LaTeX project in your LaTeX editor, and open the `chapters/chapter1.tex`.

Done? Ok let us move on then! So by now, you should have noticed a few things:

- Each sentence of the text is on a distinct line. Yet, sentences are still within the same paragraph.
- Backslash is an escape character, used at the beginning of LaTeX commands.
- To end a paragraph (or actually insert a newline), we can use the `\\` command.

Also now you know how to do a bullet list.

1.1.2 Development and importance of data privacy regulation

Chapter contain sections (defined with `\section{Name of your section}`), subsections (`\subsection`).

1.2 Explanation of 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).

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.

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

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.

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 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

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

²The 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.

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.

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

LaTeX uses 10 special characters. Each of these characters has a special meaning.

1. Ampersand (`&`) is used in tables as a cell delimiter.

2. Percent sign (%) is used for commenting a line.
3. Dollar sign (\$) is used to switch back/from mathematical notation mode.
4. Hash sign (#) is used to create macros — you definitely do not want to go more in depth here.
5. Underscore (_ or `_`) is used to indicate a subscript in maths mode, if you use it in text mode (without using backslash in front to “escape” it), your project will not compile anymore. **You may want to read that twice, and remember it.** It is in a LaTeX template, therefore it must be true.
6. Curly brackets ({ and }) or bracers are used by LaTeX commands, as you likely already noticed.
7. Tilde (~) can be used to create a non-breaking space (so that both words are on the same line).
8. Caret/Circumflex/Hat (^) is used to indicate superscript (exponent) in maths mode.
9. Backslash (\) is used in front of every command. You cannot simply escape it to print it, as `\\` create a new line.

If you happen to insert some of these symbols in your text without either escaping (when possible) or using the correct command, your project will likely not compile. Thus, you may want to be extra careful about that problem. Note: the underscore issue may also be encountered with bibliography. So if `bibtex` displays an error, it may also come from an underscore somewhere in the abstract, DOI or URL field.

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!

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

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 ** argv)
{
    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



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.

\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.

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 $0.9 \times \text{textwidth}$, 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.**

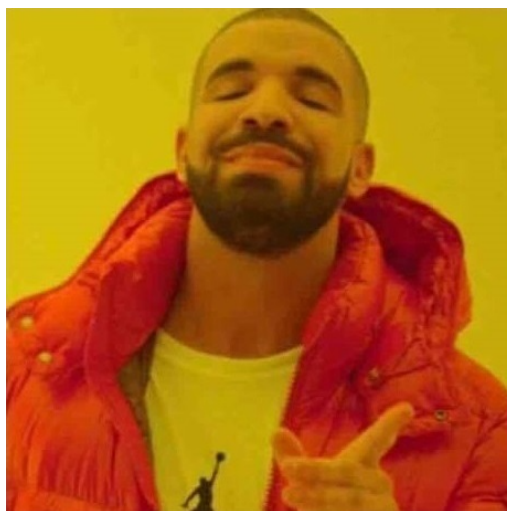
1.7.3 Multiple images in one figure

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.



Use one
single image
for the Drake
meme



Use 4 individual
images and
waste 15
minutes of my
life

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

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/>

Here is an example of confusion matrix generated:

Chest (C)	-									
Chest (ND)	*	-								
Chest (D)	*	-	*							
Ear	-	-	-							
Thigh	*	-	*	*	-					
Neck	-	*	-	*	-	-				
Palm	-		*	-	-	-	-			
Thumb	-	*	-	*	-	-	*	*		
Inner Wrist	*	-	-	-	*	*	*	-	*	
Outer Wrist	-	*	-	*	-	-	-	*	-	-
	Belly	Chest (C)	Chest (ND)	Chest (D)	Ear	Thigh	Neck	Palm	Thumb	Inner Wrist

TABLE 1.2: Post-hoc comparisons between body parts. - shows no significant difference ($p > .05$), * shows differences ($p < .05$).

Note that a table is actually a container for another type of LaTeX object, *tabular*. Tables come with captions and label, allowing us to refer to Table 1.2. Another interesting point is that the `\begin{tabular}` command uses characters. These characters specify how the text should be centered within each cell: `c` means centered, `l` means left and `r` means right. Finally, my original table was too large to fit a page, so I used the `\resizebox{\textwidth}{!}{}` command. This command needs a closing `}` after the `\end{tabular}` command. This table is also now shown in the [List of Tables](#) page.

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

1.9 Bibliography

LaTeX is really convenient to deal with bibliography. All your references should be in a `*.bib` file. Each reference has a unique key, that you will use to refer to that publication.

You can simply cite nearly anything using the `\cite` command. You

can cite conference papers, e.g. “WatchIt (Perrault et al., 2013) is an interactive wristband for smart watches.” or journal articles, e.g. “Lopez et al. (López García, 2017) ran a public consultation in Mexico”. In the first example, the key in the bib file is Perrault2013, see

```
@inproceedings{Perrault2013,
```

1.9.1 How to get Bibtex References?

The easiest way to find the Bibtex snippet you need for a given reference is to use Google Scholar ([Google Scholar n.d.](#)). On the main page, type the name of the paper you are looking for.

In the results page, locate the paper:

Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets
[ST Perrault](#), [E Lecolinet](#), [J Eagan](#)... - *Proceedings of the SIGCHI ...*, 2013 - [dl.acm.org](#)
 We present WatchIt, a prototype device that extends interaction beyond the watch surface to the wristband, and two interaction techniques for command selection and execution. Because the small screen of wristwatch computers suffers from visual occlusion and the fat finger problem, we investigated the use of the wristband as an available interaction resource. Not only does WatchIt use a cheap, energy efficient and invisible technology, but it involves simple, basic gestures that allow good performance after little training, as ...
 ☆ ⓘ Cited by 117 Related articles ⓘ

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.

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.

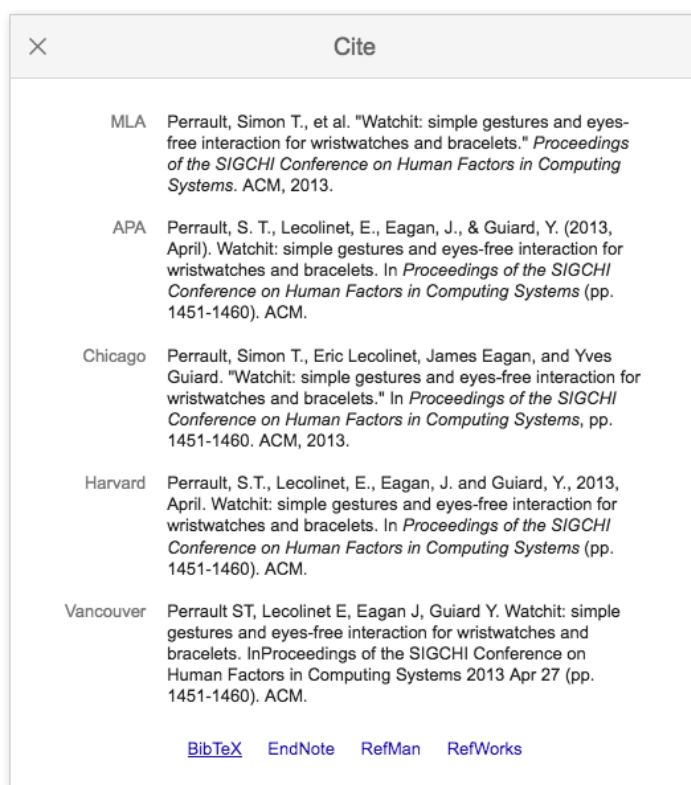


FIGURE 1.4: Pop-up window with the possible citations

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).

1.10 End of the Tips

We are now done with the tips. The next chapter contains more explanations and specificities of LaTeX and the template used here. Good luck with your capstone report.

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 L^AT_EX Thesis Template, a beautiful and easy to use template for writing a thesis using the L^AT_EX typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L^AT_EX is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L^AT_EX is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy*

mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L^AT_EX to make them look stunning.

2.3 Evaluating the effectiveness of XAI methods

L^AT_EX is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L^AT_EX is actually a simple, plain text file that contains *no formatting*. You tell L^AT_EX how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L^AT_EX is a “mark-up” language, very much like HTML.

2.3.1 User validation of explanations

If you are new to L^AT_EX, there is a very good eBook – freely available online as a PDF file – called, “The Not So Short Introduction to L^AT_EX”. The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L^AT_EX by creating several, small ‘test’ documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

2.4 What this Template Includes

2.4.1 Folders

This template comes as a single zip file that expands out to several files and folders. The folder names are mostly self-explanatory:

Appendices – this is the folder where you put the appendices. Each appendix should go into its own separate `.tex` file. An example and template are included in the directory.

Chapters – this is the folder where you put the thesis chapters. A thesis usually has about six chapters, though there is no hard rule on this. Each chapter should go in its own separate `.tex` file and they can be split as:

- Chapter 1: Introduction to the thesis topic
- Chapter 2: Background information and theory
- Chapter 3: (Laboratory) experimental setup
- Chapter 4: Details of experiment 1
- Chapter 5: Details of experiment 2
- Chapter 6: Discussion of the experimental results
- Chapter 7: Conclusion and future directions

This chapter layout is specialised for the experimental sciences, your discipline may be different.

Figures – this folder contains all figures for the thesis. These are the final images that will go into the thesis document.

2.4.2 Files

Included are also several files, most of them are plain text and you can see their contents in a text editor. After initial compilation, you will see that more auxiliary files are created by \LaTeX or BibTeX and which you don't need to delete or worry about:

example.bib – this is an important file that contains all the bibliographic information and references that you will be citing in the thesis for use with BibTeX. You can write it manually, but there are reference manager programs available that will create and manage it for you. Bibliographies in \LaTeX are a large subject and you may need to read about BibTeX before starting with this. Many modern reference managers will allow you to export your references in BibTeX format which greatly eases the amount of work you have to do.

MastersDoctoralThesis.cls – this is an important file. It is the class file that tells \LaTeX how to format the thesis.

main.pdf – this is your beautifully typeset thesis (in the PDF file format) created by \LaTeX . It is supplied in the PDF with the template and after you compile the template you should get an identical version.

main.tex – this is an important file. This is the file that you tell \LaTeX to compile to produce your thesis as a PDF file. It contains the framework

and constructs that tell \LaTeX how to layout the thesis. It is heavily commented so you can read exactly what each line of code does and why it is there. After you put your own information into the *THESIS INFORMATION* block – you have now started your thesis!

Files that are *not* included, but are created by \LaTeX as auxiliary files include:

main.aux – this is an auxiliary file generated by \LaTeX , if it is deleted \LaTeX simply regenerates it when you run the main .tex file.

main.bbl – this is an auxiliary file generated by BibTeX, if it is deleted, BibTeX simply regenerates it when you run the main.aux file. Whereas the .bib file contains all the references you have, this .bbl file contains the references you have actually cited in the thesis and is used to build the bibliography section of the thesis.

main.blg – this is an auxiliary file generated by BibTeX, if it is deleted BibTeX simply regenerates it when you run the main .aux file.

main.lof – this is an auxiliary file generated by \LaTeX , if it is deleted \LaTeX simply regenerates it when you run the main .tex file. It tells \LaTeX how to build the *List of Figures* section.

main.log – this is an auxiliary file generated by \LaTeX , if it is deleted \LaTeX simply regenerates it when you run the main .tex file. It contains messages from \LaTeX , if you receive errors and warnings from \LaTeX , they will be in this .log file.

main.lot – this is an auxiliary file generated by \LaTeX , if it is deleted \LaTeX simply regenerates it when you run the main .tex file. It tells \LaTeX how to build the *List of Tables* section.

main.out – this is an auxiliary file generated by L^AT_EX, if it is deleted L^AT_EX simply regenerates it when you run the main .tex file.

So from this long list, only the files with the .bib, .cls and .tex extensions are the most important ones. The other auxiliary files can be ignored or deleted as L^AT_EX and BibTeX will regenerate them.

Finally, there is the block where the chapters are included. Uncomment the lines (delete the % character) as you write the chapters. Each chapter should be written in its own file and put into the *Chapters* folder and named Chapter1, Chapter2, etc... Similarly for the appendices, uncomment the lines as you need them. Each appendix should go into its own file and placed in the *Appendices* folder.

After the preamble, chapters and appendices finally comes the bibliography. The bibliography style (called *authoryear*) is used for the bibliography and is a fully featured style that will even include links to where the referenced paper can be found online. Do not underestimate how grateful your reader will be to find that a reference to a paper is just a click away. Of course, this relies on you putting the URL information into the BibTeX file in the first place.

2.5 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.

2.5.1 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 L^AT_EX 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).

A Note on `bibtex`

The `bibtex` backend used in the template by default does not correctly handle unicode character encoding (i.e. "international" characters). You may see a warning about this in the compilation log and, if your references contain unicode characters, they may not show up correctly or at all. The solution to this is to use the `biber` backend instead of the outdated

bibtex backend. This is done by finding this in `main.tex`: `backend=bibtex` and changing it to `backend=biber`. You will then need to delete all auxiliary BibTeX files and navigate to the template directory in your terminal (command prompt). Once there, simply type `biber main` and biber will compile your bibliography. You can then compile `main.tex` as normal and your bibliography will be updated. An alternative is to set up your LaTeX editor to compile with biber instead of bibtex, see [here](#) for how to do this for various editors.

2.5.2 Tables

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

```
\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
\thead{Groups} & \thead{Treatment X} & \thead{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}
```

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

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).

2.5.3 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



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

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.

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.

L^AT_EX is capable of using images in pdf, jpg and png format.

2.6 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. L^AT_EX 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`.

2.7 In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own `Chapter1.tex` and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —

Sunil Patel: www.sunilpatel.co.uk

Vel: LaTeXTemplates.com

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 `.yaml` format, with one `.yaml` 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 `.yaml` 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 as 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 networks², 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)

²State 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 $3 \leq N \leq 10$) 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.

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

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

Google Scholar (n.d.). <http://scholar.google.com.sg/>. Last Accessed: 1929-10-24.

López García, David (2017). "Mediation Styles and Participants' Perception of Success in Consultative Councils: The case of Guadalajara, Mexico". In: *Journal of Public Deliberation* 13.2, p. 10.

Perrault, Simon T et al. (2013). "Watchit: simple gestures and eyes-free interaction for wristwatches and bracelets". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1451–1460.

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}.
```