Faculty of Engineering
Office of Undergraduate Programmes
EG3602/EG3612 Vacation Internship Programme

NUS
National University of Singapore

VIP REPORT CLEARANCE FORM

Please ensure that this form is attached at the back of your VIP report before the submission to the faculty.

Student Information

Name of Student: He Cheng Hui

Department: Engineering Science Programme

Matriculation No: A0190274E

Report: End-Term

Company Information

Name of Company: Ministry of Home Affairs, Home Team Science and Technology Agency (HTX)

Name of Supervisor: Ng Gee Wah

Designation: Director Trial and Experimentation @ HTX

Email / Telephone: 6478 7530

Report Clearance by Company

Signature	Company Stamp	Date
S G.W.	Home Team Science And Technology Agency (HTX) Ministry of Home Affairs	11th August 2020

Note: If the company wishes to have a copy of the report, the arrangement is left between the company and the intern.

Faculty Internship Team
Email: enginternship@nus.edu.sg

National University of Singapore

Faculty of Engineering

Module	☐ EG3611 (IA) ☐ EG1611 (Co-		☐ EG3611A (IA) ☐ EG2620 (Co-	☒ EG3612
Report	☐ Interim		☒ Final	
Reporting Period	11/5/2020	to	7/8/2020	

Name	**HE CHENG HUI**
Matriculation	**A0190274E**
Department	**Engineering Science**
Internship Period	**11th May 2020 – 7th August 2020**
Mentor	**Karl Erik Birgersson**
Company	**Ministry of Home Affairs - Home Team Science and Technology Agency**

Table of Contents

Acknowledgements

I would like to express my appreciation to all those who provided me the possibility to complete this internship. I would like to express my deepest gratitude to my intern supervisor, Dr Ng, who gave me the opportunity and providing necessary help to complete the projects during the internship.

I would like to express my gratitude to Mr Tan, who helped us bridge the communication between the interns and the procurement team.

I would like to also express my gratitude to Mr Leung, who helped take care of all our administrative items.

A special thanks goes to Mr Goh, who helped answer my coding questions.

Introduction

Home Team Science and Technology Agency (HTX) is one of the agencies under Ministry of Home Affairs (MHA). Its purpose is to bring together science and engineering capabilities to better improve the landscape of our homeland security. HTX's goal is to facilitate greater co-operation and synergy across the entire homeland security ecosystem and enable the Home Team departments to adopt a unified approach to protect Singapore.

To meet Home Teams' unique requirements to operate in complex environments to counter technologically fuelled threats, HTX focus on 13 key areas of development. They are Biometrics and Profiling, C4I, CBRNE, Cybersecurity, Data Science and AI, Digital and Information Forensics, Forensics, Human Factors and Simulation, Land Systems, Marine Systems, Protective Security and Safety, Robotics, Automation and Unmanned Systems (RAUS) and finally Sense-making and Surveillance. During my internship period, I was under the Trails and Experimentation department.

In this report, I will discuss the two projects that I was involved in. The first project is on text analytics using a natural language tool called spaCy, while the second project is on the generation of E-learning material using Text-to-Speech (TTS).

Project 1 – Text analytics Using spaCy

Natural Language Processing (NLP) is a way to programme computers to process and analyse human language. The NLP tool used in this project is called spaCy, which is an open-source software library for advanced NLP in Python, recommended by Dr Ng. spaCy is advertised as an “industrial strength” NLP tool, which means that it is one of the most powerful software libraries for computers to understand human language [1].

In the beginning, spaCy was proposed to be used alongside Excel to help identify similar text among the thousands of rows of data. By using the similarity function built into spaCy, one can calculate the cosine similarity between two text after running both text through its NLP pipeline as seen below [2].

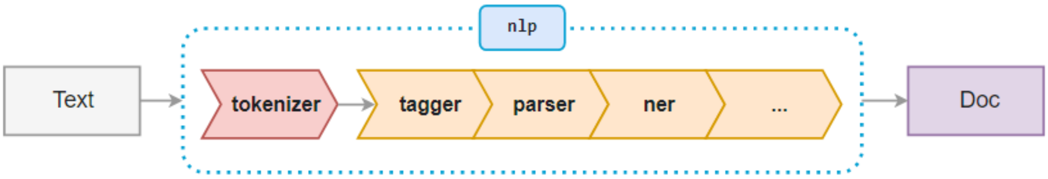
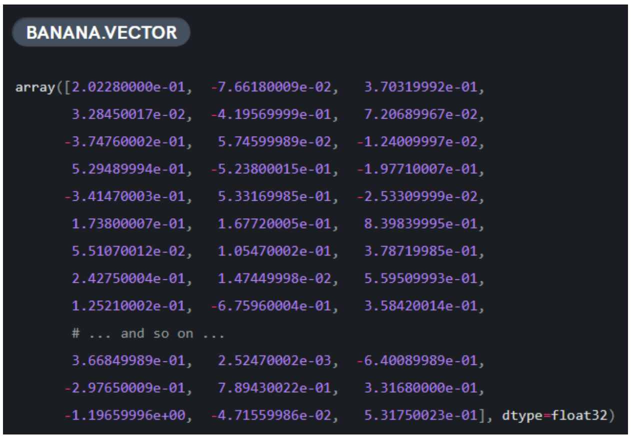


Figure 1: NLP pipeline of a default spaCy model

When “nlp” is called on a text, spaCy first tokenises them by segmenting words into tokens, to produce a Doc object. The NLP pipeline differs base on the type of model one loads into spaCy, and the pipeline seen in Figure 1 is using the default models (small, medium or large models) that spaCy provides. In the default pipeline, the “tagger” assign part-of-speech tags, the “parser” assign dependency labels and the “ner” detect and label named entities. Furthermore, users can also create and add custom components to the pipeline. Before finding out the similarity, each word will be assigned a vector that is generated using an algorithm called the Word2Vec. Each word vectors is a multi-dimensional meaning representations of a word as seen below that shows the word vectors of the word “Banana” [2].



```
BANANA.VECTOR

array([2.02280000e-01,  -7.66180009e-02,   3.70319992e-01,
        3.28450017e-02,  -4.19569999e-01,   7.20689967e-02,
       -3.74760002e-01,   5.74599989e-02,  -1.24009997e-02,
        5.29489994e-01,  -5.23800015e-01,  -1.97710007e-01,
       -3.41470003e-01,   5.33169985e-01,  -2.53309999e-02,
        1.73800007e-01,   1.67720005e-01,   8.39839995e-01,
        5.51070012e-02,   1.05470002e-01,   3.78719985e-01,
        2.42750004e-01,   1.47449998e-02,   5.59509993e-01,
        1.25210002e-01,  -6.75960004e-01,   3.58420014e-01,
        # ... and so on ...
        3.66849989e-01,   2.52470002e-03,  -6.40089989e-01,
       -2.97650009e-01,   7.89430022e-01,   3.31680000e-01,
       -1.19659996e+00,  -4.71559986e-02,   5.31750023e-01], dtype=float32)
```

Figure 2: Word vectors representation of "Banana" generated by Word2Vec

The first challenge I encountered was that the models doesn’t work well with short forms and some local terms. This is because the models are trained on text from blogs, news and comments around the internet. Since having to train a new model for our use case was deemed too complex and possibly time consuming, pre-processing was done on text first, to let the Word2Vec algorithm perform better. This was done by first transforming each character into its lowercase form, remove any stop words, punctuations and then lemmatise each word. Furthermore, I was able to decide which model to use moving forward, based on the initial test results. The full analysis can be found in Appendix A, Figures A1 and A2. From Figures A1 and A2, we compare medium and large default models because the small default model does not have enough information to calculate the similarity score between two text. The outliers found in Figure A2 refers to the test cases that spaCy computed to have low similarity scores, even though they are the same text with different wordings. Based on the data present in both Figures, there is an average difference of 1.552689015% between medium and large model's similarity score. On the other hand, there is an average difference of 42.28285579% between medium and large model's computation time. Therefore, all future spaCy programmes were done using the default large model due to its lower computational cost while having relatively the same output as the medium default model.

The second challenge I encountered was the software limitations of the Whole of Government (WOG) laptops. Due to security issues, I was unable to install Python directly into the laptops. Hence, I was tasked to try package my programme as an executable, as it might be easier to get approval. However, I was unable to do so even after two weeks of testing. Nonetheless, Dr Ng was still determined to find another use for spaCy and thus, it got repurposed as a standalone text analytics tool for news articles.

The purpose of the text analytics tool for news articles is to suggest to the user, how similar or different two news articles are. The main challenge here is to set accurate limits that separates the different categories of similarity. In the programme I made, similarity results were split into three categories; similar, some similarities and not similar. Which result appearing would be based on the similarity score output by spaCy. During initial testing of the programme, I found that my perception of similarity and a computer's version of similarity is different. This is because my perception of how similar two text are, is based on the semantic similarity between them, whereas spaCy perceive similarity based on the Word2Vec vectors between them. Hence, if two text have high amount of similar words but no contextual similarity, spaCy could output a high similarity score. However, after discussing with Dr Ng, we decided to go ahead and use the values I got from my initial testing. The test cases can be seen in Appendix A, Figures A3 to A5. The challenge of setting an accurate similarity score limit for the correct similarity result was mainly the lack of time to get enough and properly classifies test cases.

After getting the main code and limit done and approved, the next step is to create a Graphical User Interface (GUI) for users to load multiple news article URLs and even their own text documents to compare. Continuing the trend of using Python, the GUI was created using the PyQt5 library. Here, another challenge arises as PyQt uses object-oriented programming and their own design framework, which was new to me. However, with the help of Mr Goh, I was able to integrate spaCy into an engineering GUI seen below.

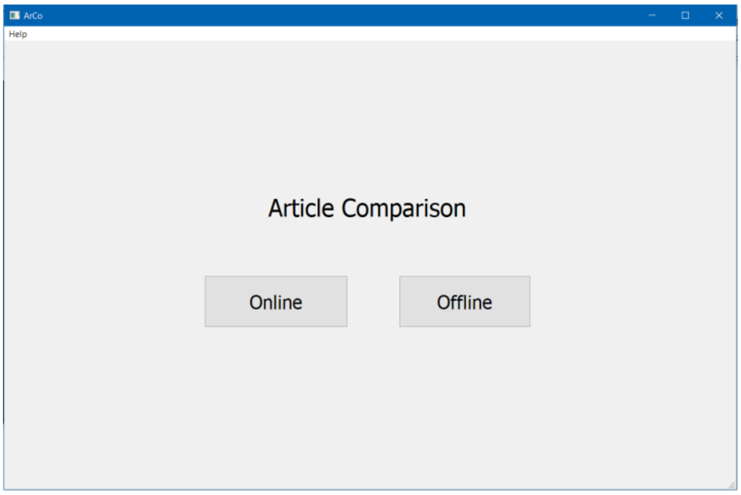
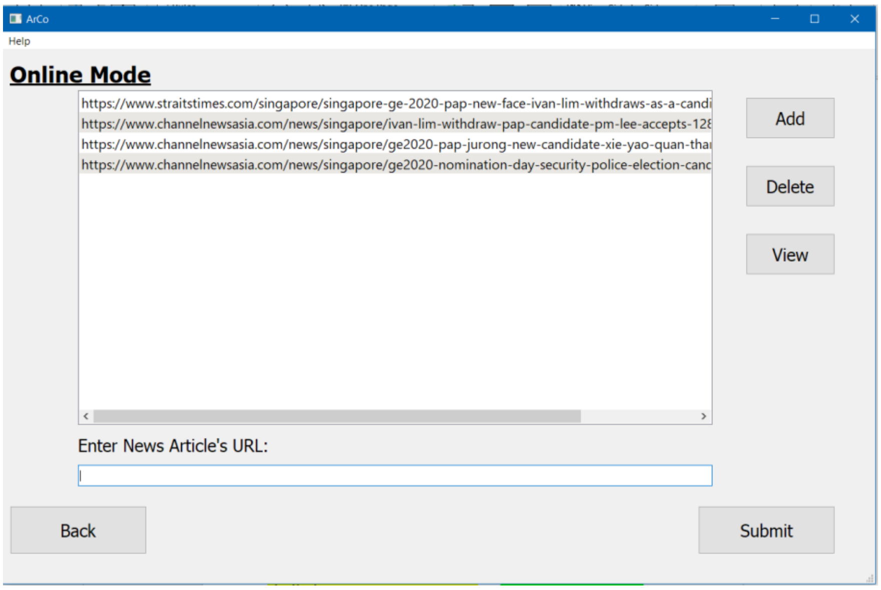


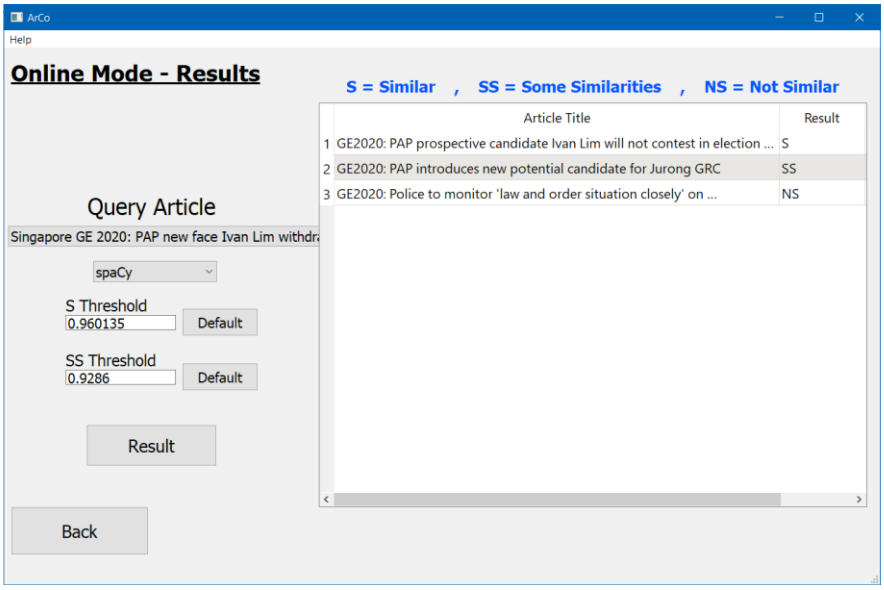
Figure 3: Main page of the engineering GUI with two selectable modes

As seen in Figure 3, the GUI has two main mode, online and offline. The online mode is able to take in URL inputs from news sites and then compare any selected article to the rest of the URL pool as seen in Figure 4 and 5 below.



https://www.straitstimes.com/singapore/singapore-ge-2020-pap-new-face-ivan-lim-withdraws-as-a-candi
https://www.channelnewsasia.com/news/singapore/ivan-lim-withdraw-pap-candidate-pm-lee-accepts-128
https://www.channelnewsasia.com/news/singapore/ge2020-pap-jurong-new-candidate-xie-yao-quan-thai
https://www.channelnewsasia.com/news/singapore/ge2020-nomination-day-security-police-election-canc

Figure 4: URL input page where users can either enter, delete or even view the URL that they entered



	Article Title	Result
1	GE2020: PAP prospective candidate Ivan Lim will not contest in election ...	S
2	GE2020: PAP introduces new potential candidate for Jurong GRC	SS
3	GE2020: Police to monitor 'law and order situation closely' on ...	NS

Singapore GE 2020: PAP new face Ivan Lim withdr

spaCy

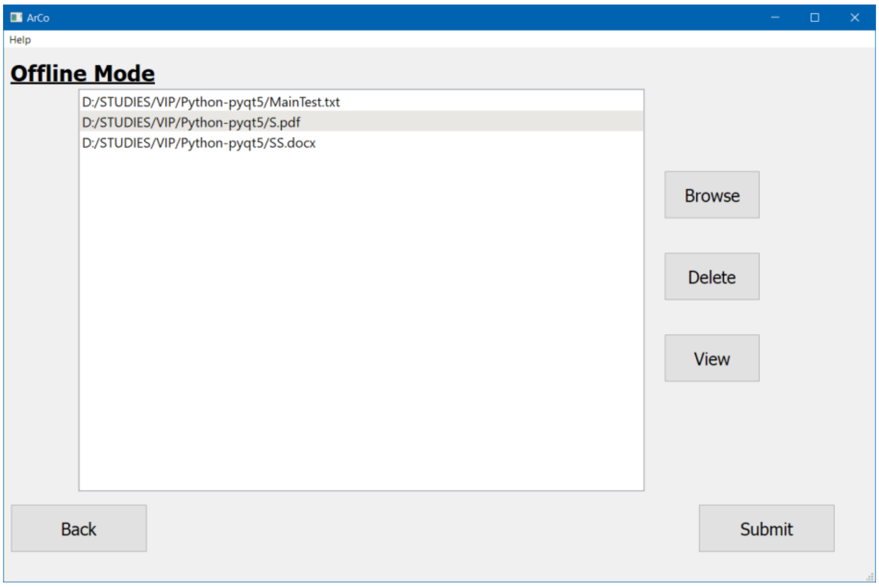
S Threshold
0.960135

SS Threshold
0.9286

Figure 5: Online mode results page where users can choose the main article to compare against the rest of the pool on the left. Results are shown on the right.

As seen in Figure 4, users can enter URLs from news sites of their choosing. The input page also comes with a delete function and a view function if the user wishes to go to the URL in their browser. One caveat of this section is that it is not able to detect if the URL is from a news site, though the programme would still work. In Figure 5, users can select which article to test against in the dropdown box under the "Query Article" label. Clicking on the [Results] button would pass the scraped news article from the URL into spaCy, and the result would be shown on the right, under the "Results" column. As seen above in blue, there are three different outcomes; S, SS and NS. Not shown in Figure 5 is the similarity score that is to the right of the "Results" column. It was purposely hidden as the values would only serve to confuse the user. Nonetheless, users can still scroll to the right to check the similarity score, and fine tune the results by changing the value under the "S" and "SS threshold" on the left.

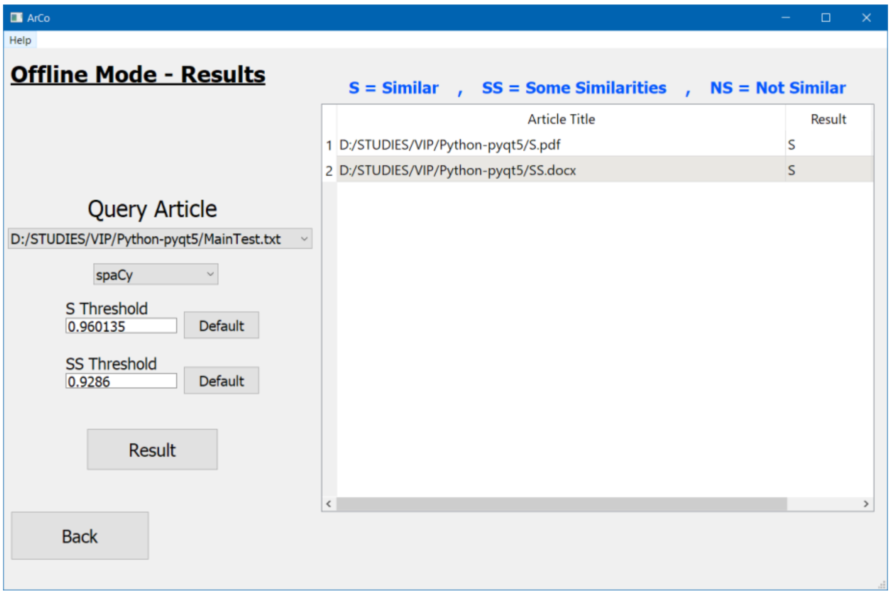
The offline mode on the other hand, would require the user to import their own files from their computer directories, as seen below.



Offline Mode

D:/STUDIES/VIP/Python-pyqt5/MainTest.txt
D:/STUDIES/VIP/Python-pyqt5/S.pdf
D:/STUDIES/VIP/Python-pyqt5/SS.docx

Figure 6: File import page where users can either upload, delete or view the files they uploaded



Offline Mode - Results

S = Similar , SS = Some Similarities , NS = Not Similar

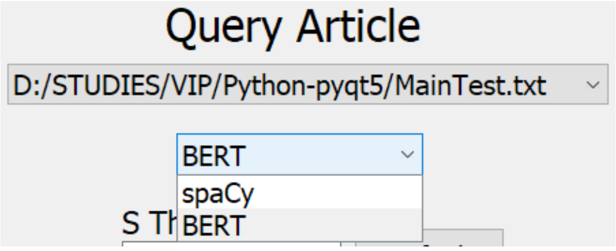
	Article Title	Result
1	D:/STUDIES/VIP/Python-pyqt5/S.pdf	S
2	D:/STUDIES/VIP/Python-pyqt5/SS.docx	S

Query Article
D:/STUDIES/VIP/Python-pyqt5/MainTest.txt
spaCy
S Threshold 0.960135
SS Threshold 0.9286

Figure 7: Offline results page where users can choose the main article to compare against the rest of the pool on the left. Results are shown on the right.

As seen in Figure 6, offline mode refers to users importing their own files to the programme. In this engineering GUI, it can accept text, pdf, and Microsoft word files as those are some of the most common document types. In Figure 7, the structure and the article uploaded is the same as in Figure 5. However, another caveat would be the difference in results when comparing the outcomes of articles from URLs and the same articles when it is uploaded. This is because in online mode, the news scraping library used to extract the articles, also included extra words like "advertisements" that served as placeholder for the actual advertisement one sees on the actual website. These words were another challenge for me, as different news sites would have different layouts, which means the news scraping library would include different extra words for different news sites entered. Due to the lack of time, I was not able to figure out to remove those extra words.

Another feature of the engineering GUI is the option to choose between two models; spaCy's default large model and spaCy – BERT model as seen below.



Query Article

D:/STUDIES/VIP/Python-pyqt5/MainTest.txt

Figure 8: Option to choose which model to use when in the results page

BERT is an acronym for "Bidirectional Encoder Representations from Transformers". In summary, it is the state-of-the-art NLP language model that looks at both left and right side of a word when it was trained [3]. Luckily, spaCy published an open-source wrapper library called "spacy – transformers" that allowed me to use BERT in spaCy's framework. This library basically replaces the default NLP pipeline as seen in Figure 1 to its own pipeline as seen below.

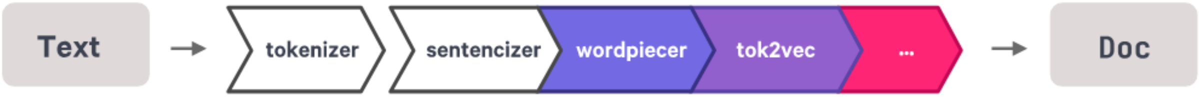
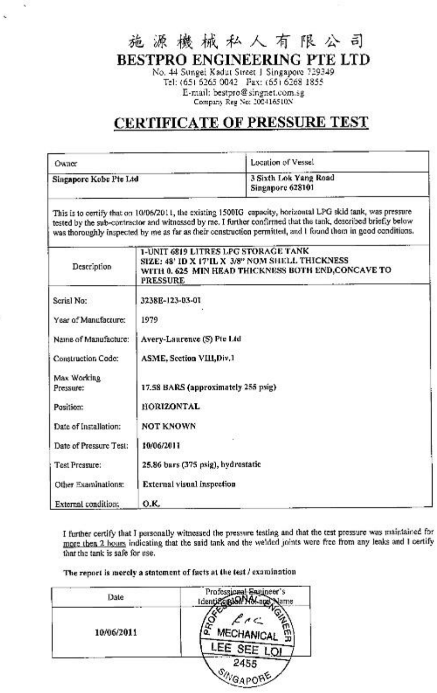


Figure 9: Custom NLP pipeline when using spaCy -BERT model

As seen in Figure 9, after tokenising, it splits the text by sentences, then performs wordpiece pre-processing before running the transformer over the document with tok2vec [4]. Due to the "sentenciser", I remove the pre-processing function seen in the default model, since that would technically ruin BERT's understanding of each sentence. Initial testing of this model can be seen in Appendix A, Figures A3 to A5.

Another possible feature of this engineering GUI is to use it as a pdf document checker as proposed by Dr Ng. The idea is to extract the text in the scanned images of a pdf document, then check it contains the correct signatures, as seen below.



施 源 機 械 私 人 有 限 公 司
BESTPRO ENGINEERING PTE LTD
No. 44 Sungei Kadut Street 1 Singapore 729349
Tel: (65) 6265 0042 Fax: (65) 6268 1855
E-mail: bestpro@singnet.com.sg
Company Reg No: 200416510N

CERTIFICATE OF PRESSURE TEST

Owner	Location of Vessel
Singapore Kobe Pte Ltd	**3 Sixth Lok Yang Road Singapore 628101**

This is to certify that on 10/06/2011, the existing 1500IG capacity, horizontal LPG skid tank, was pressure tested by the sub-contractor and witnessed by me. I further confirmed that the tank, described briefly below was thoroughly inspected by me as far as their construction permitted, and I found them in good conditions.

Description	**1-UNIT 6819 LITRES LPG STORAGE TANK SIZE: 48' ID X 17'IL X 3/8" NOM SHELL THICKNESS WITH 0. 625 MIN HEAD THICKNESS BOTH END,CONCAVE TO PRESSURE**
Serial No:	**3238E-123-03-01**
Year of Manufacture:	1979
Name of Manufacture:	**Avery-Laurence (S) Pte Ltd**
Construction Code:	**ASME, Section VIII,Div.1**
Max Working Pressure:	**17.58 BARS (approximately 255 psig)**
Position:	**HORIZONTAL**
Date of Installation:	**NOT KNOWN**
Date of Pressure Test:	**10/06/2011**
Test Pressure:	**25.86 bars (375 psig), hydrostatic**
Other Examinations:	**External visual inspection**
External condition:	**O.K.**

I further certify that I personally witnessed the pressure testing and that the test pressure was maintained for more then 2 hours indicating that the said tank and the welded joints were free from any leaks and I certify that the tank is safe for use.

The report is merely a statement of facts at the test / examination

Date	Professional Engineer's Identification No. and Name
10/06/2011	PROFESSIONAL ENGINEER MECHANICAL LEE SEE LOI 2455 SINGAPORE

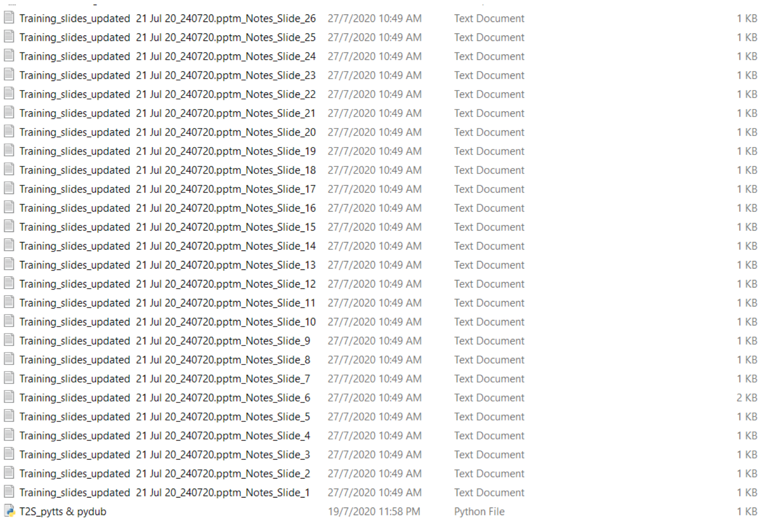
Figure 10: Example of a page of a scanned pdf document to be checked

As seen in Figure 10, it shows an example of a page to check. Using a Python computer vision library called "pytesseract", I managed to extract out most of the text of page but not the stamp at the bottom, which is the most important part to check. Unfortunately, due to the complexity of computer vision pre-processing techniques and the lack of time, I was unable to optimise it further then a surface implementation. This includes implementing the technique for purely text pdf documents as well, in case of an event where both text and scanned images are present in the pdf document.

Project 2 – E-learning using TTS

A side project that I was tasked with is to produce e-learning materials for a department called the “Procurement Team”. The distribution format that the team chose is to have a PowerPoint slideshow with audio. PowerPoint was chosen because they want to include interactive quizzes to make the lesson more engaging. Hence, the method I used to automate the creation of audio e-learning material given the scripts is through a Python library called “Pyttsx3”. It is an open source text-to-speech (TTS) library that works offline and is able to change voice type, volume and talking rate. Originally, another Python TTS library called “gTTS” (Google TTS) was used, but users’ feedback not liking the sound of the audio output.

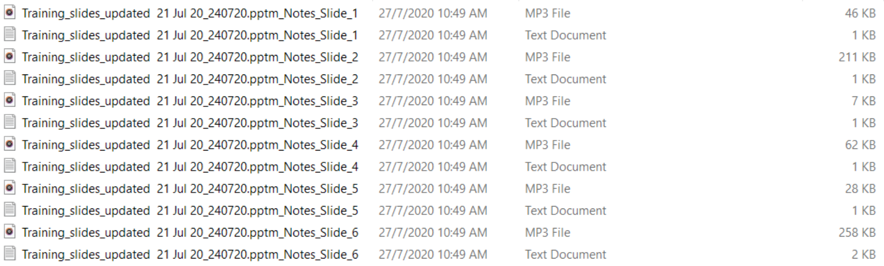
The workflow will first start with generating the text files of the scripts so the Python script can output their corresponding audio files as seen below. This can be done by first running a macro code in the PowerPoint that has the script already written in the presenter’s note section. The macro code can be found in Appendix B, Figure B1.



Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_26	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_25	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_24	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_23	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_22	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_21	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_20	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_19	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_18	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_17	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_16	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_15	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_14	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_13	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_12	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_11	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_10	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_9	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_8	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_7	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_6	27/7/2020 10:49 AM	Text Document	2 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_5	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_4	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_3	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_2	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_1	27/7/2020 10:49 AM	Text Document	1 KB
T2S_pytts & pydub	19/7/2020 11:58 PM	Python File	1 KB

Figure 11: Text files generated after running a specific macro code in PowerPoint

As seen in Figure 11, each text file will be named after the PowerPoint file name and the slide number where the script is stored. Then, running the Python file would generate the audio files as seen below.



Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_1	27/7/2020 10:49 AM	MP3 File	46 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_1	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_2	27/7/2020 10:49 AM	MP3 File	211 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_2	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_3	27/7/2020 10:49 AM	MP3 File	7 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_3	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_4	27/7/2020 10:49 AM	MP3 File	62 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_4	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_5	27/7/2020 10:49 AM	MP3 File	28 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_5	27/7/2020 10:49 AM	Text Document	1 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_6	27/7/2020 10:49 AM	MP3 File	258 KB
Training_slides_updated 21 Jul 20_240720.pptm_Notes_Slide_6	27/7/2020 10:49 AM	Text Document	2 KB

Figure 12: e-learning audio materials generated from each text files

As seen in Figure 12, each audio file will have the same naming convention as the text file it got its input from. Hence, making it easy for the user to choose which audio to insert into the final PowerPoint and for easier editing.

Conclusion

Being a student in Engineering Science pursuing a specialisation in Computer Engineering Science, I am very satisfied with the internship as it gave me a valuable chance to work with Machine Learning, NLP, TTS and Computer Vision, with real life problems. The physical and virtual tech sharing with HTX personals at the end of my internship period also gave me a rare opportunity to interact with people from different technological departments. This gave me many insights into how one technology can be used for so many other purposes. Even though the internship was over, it gave me joy to know that my hard work over this summer would continue to be developed by other people in HTX.

References

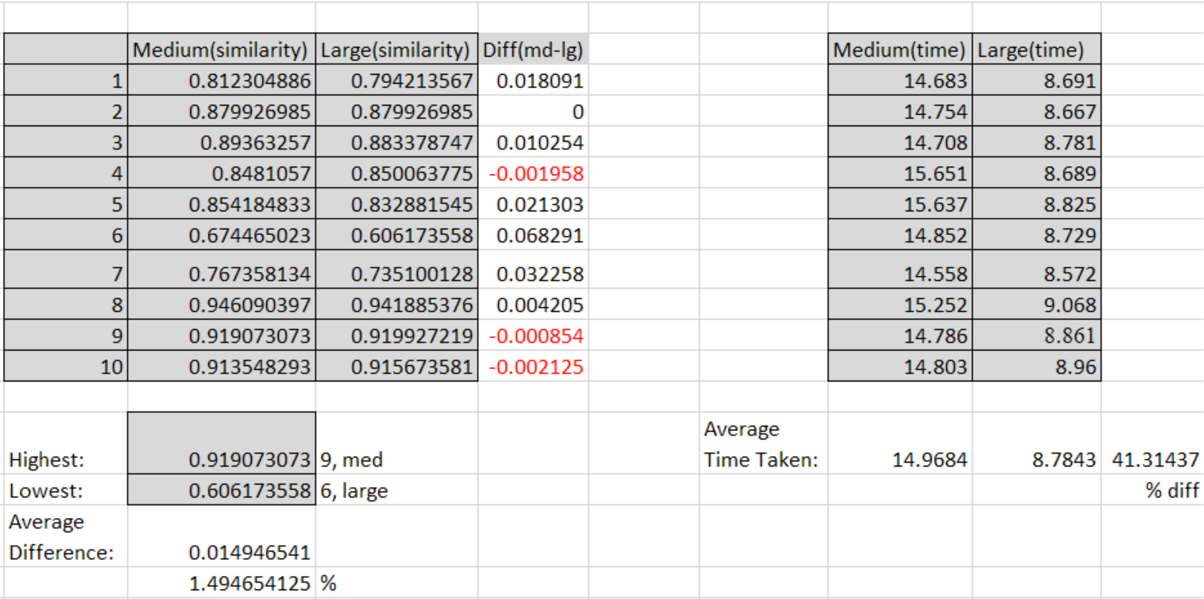
1. McDonald, C. A short introduction to NLP in Python with spaCy. Medium (2020). at https://towardsdatascience.com/a-short-introduction-to-nlp-in-python-with-spacy-d0aa819af3ad

2. Spacy.io (2020). at https://spacy.io/usage/processing-pipelines

3. Horev, R. BERT Explained: State of the art language model for NLP. Medium (2020). at https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

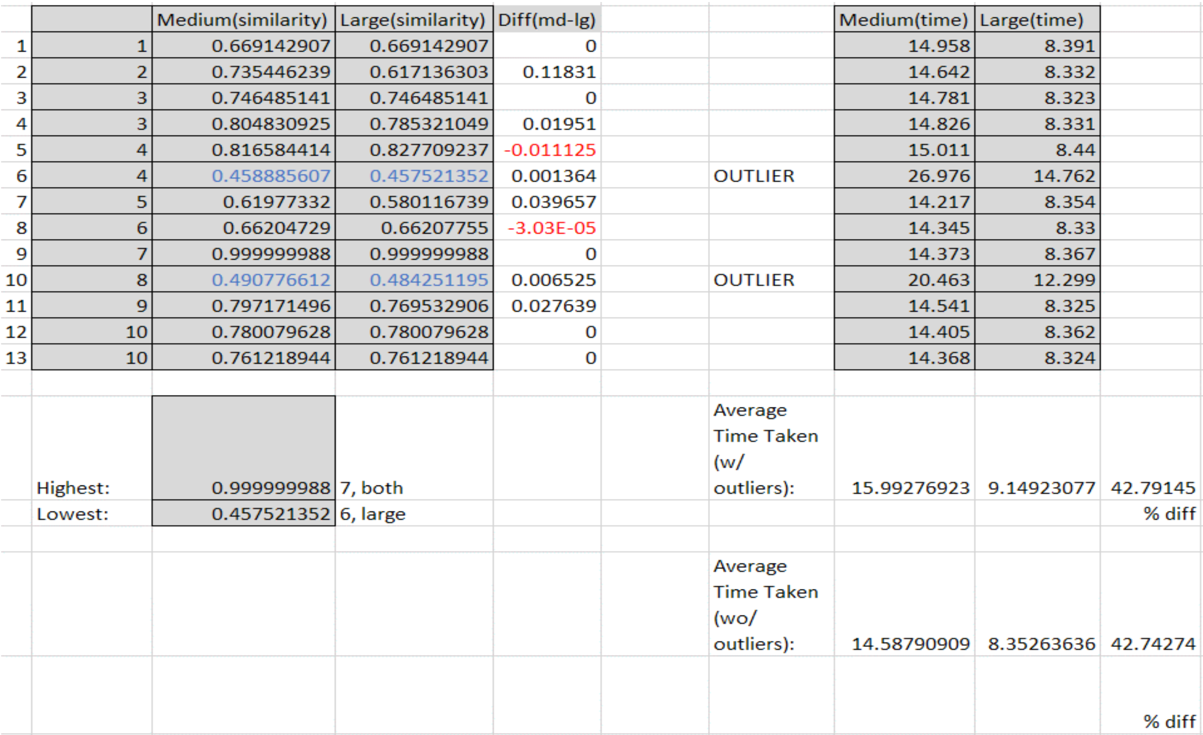
4. HONNIBAL, M. & MONTANI, I. spaCy meets Transformers: Fine-tune BERT, XLNet and GPT-2 · Explosion. Explosion (2020). at https://explosion.ai/blog/spacy-transformers

Appendix A : Testing Results



	Medium(similarity)	Large(similarity)	Diff(md-lg)			Medium(time)	Large(time)	
1	0.812304886	0.794213567	0.018091			14.683	8.691	
2	0.879926985	0.879926985	0			14.754	8.667	
3	0.89363257	0.883378747	0.010254			14.708	8.781	
4	0.8481057	0.850063775	-0.001958			15.651	8.689	
5	0.854184833	0.832881545	0.021303			15.637	8.825	
6	0.674465023	0.606173558	0.068291			14.852	8.729	
7	0.767358134	0.735100128	0.032258			14.558	8.572	
8	0.946090397	0.941885376	0.004205			15.252	9.068	
9	0.919073073	0.919927219	-0.000854			14.786	8.861	
10	0.913548293	0.915673581	-0.002125			14.803	8.96	
Highest:	0.919073073	9, med			Average Time Taken:	14.9684	8.7843	41.31437
Lowest:	0.606173558	6, large						% diff
Average Difference:	0.014946541							
	1.494654125	%						

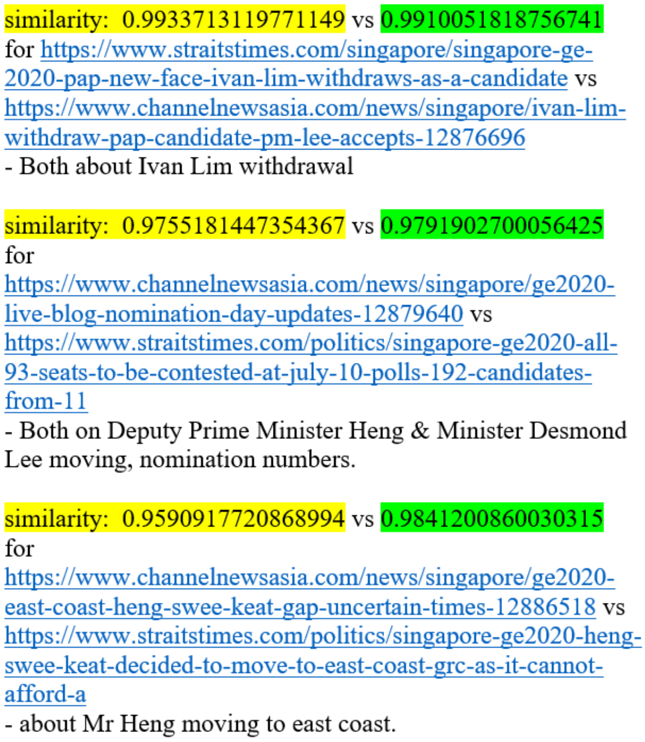
Figure A1: Time taken for Medium and Large default models when calculating similarity score between 2 different sources



		Medium(similarity)	Large(similarity)	Diff(md-lg)			Medium(time)	Large(time)	
1	1	0.669142907	0.669142907	0			14.958	8.391	
2	2	0.735446239	0.617136303	0.11831			14.642	8.332	
3	3	0.746485141	0.746485141	0			14.781	8.323	
4	3	0.804830925	0.785321049	0.01951			14.826	8.331	
5	4	0.816584414	0.827709237	-0.011125			15.011	8.44	
6	4	0.458885607	0.457521352	0.001364		OUTLIER	26.976	14.762	
7	5	0.61977332	0.580116739	0.039657			14.217	8.354	
8	6	0.66204729	0.66207755	-3.03E-05			14.345	8.33	
9	7	0.999999988	0.999999988	0			14.373	8.367	
10	8	0.490776612	0.484251195	0.006525		OUTLIER	20.463	12.299	
11	9	0.797171496	0.769532906	0.027639			14.541	8.325	
12	10	0.780079628	0.780079628	0			14.405	8.362	
13	10	0.761218944	0.761218944	0			14.368	8.324	
	Highest:	0.999999988	7, both			Average Time Taken (w/ outliers):	15.99276923	9.14923077	42.79145
	Lowest:	0.457521352	6, large						% diff
						Average Time Taken (wo/ outliers):	14.58790909	8.35263636	42.74274
									% diff

Figure A2: Time taken for Medium and Large default models when calculating similarity score between another 2 different sources, with and without the presence of outliers

Case 1: Similar reports

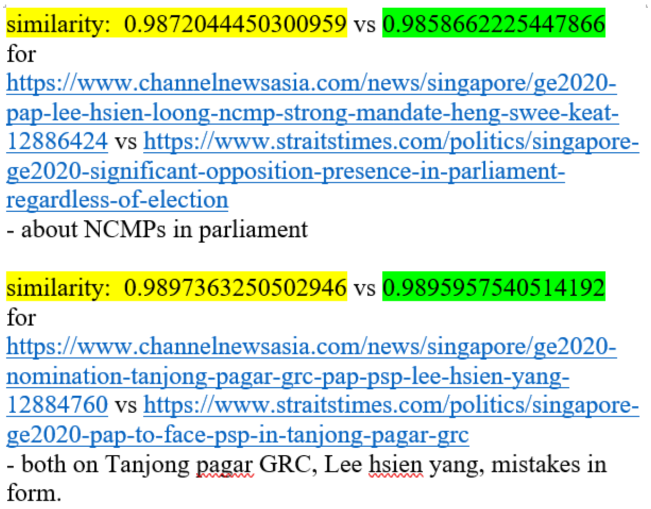


similarity: 0.9933713119771149 vs 0.9910051818756741
for https://www.straitstimes.com/singapore/singapore-ge-2020-pap-new-face-ivan-lim-withdraws-as-a-candidate vs https://www.channelnewsasia.com/news/singapore/ivan-lim-withdraw-pap-candidate-pm-lee-accepts-12876696
- Both about Ivan Lim withdrawal

similarity: 0.9755181447354367 vs 0.9791902700056425
for
https://www.channelnewsasia.com/news/singapore/ge2020-live-blog-nomination-day-updates-12879640 vs https://www.straitstimes.com/politics/singapore-ge2020-all-93-seats-to-be-contested-at-july-10-polls-192-candidates-from-11
- Both on Deputy Prime Minister Heng & Minister Desmond Lee moving, nomination numbers.

similarity: 0.9590917720868994 vs 0.9841200860030315
for
https://www.channelnewsasia.com/news/singapore/ge2020-east-coast-heng-swee-keat-gap-uncertain-times-12886518 vs https://www.straitstimes.com/politics/singapore-ge2020-heng-swee-keat-decided-to-move-to-east-coast-grc-as-it-cannot-afford-a
- about Mr Heng moving to east coast.

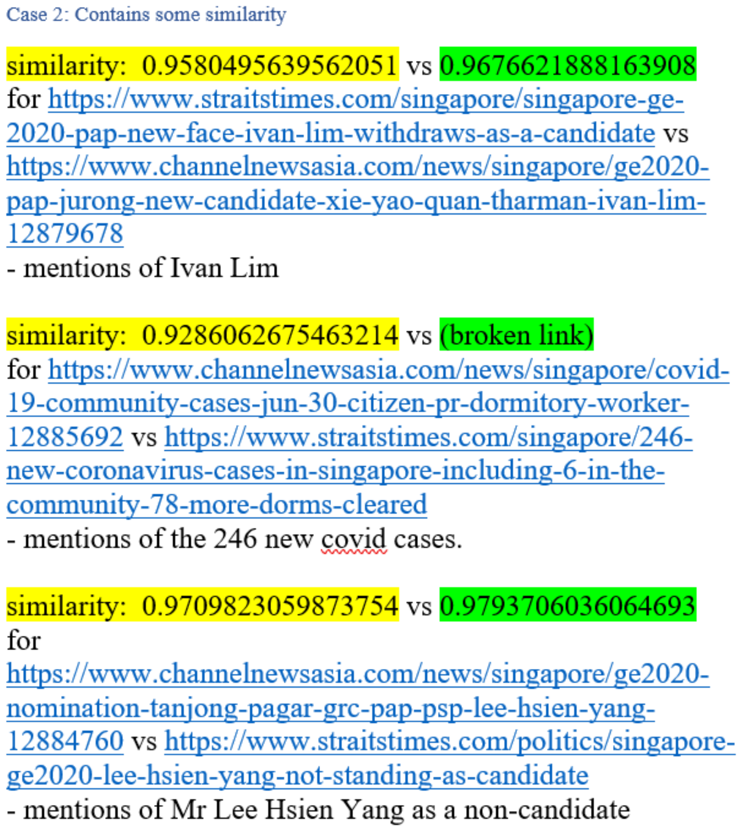
Figure A3a: Pairs of self-sourced and self-classified similar news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model



similarity: 0.9872044450300959 vs 0.9858662225447866
for
https://www.channelnewsasia.com/news/singapore/ge2020-pap-lee-hsien-loong-ncmp-strong-mandate-heng-swee-keat-12886424 vs https://www.straitstimes.com/politics/singapore-ge2020-significant-opposition-presence-in-parliament-regardless-of-election
- about NCMPs in parliament

similarity: 0.9897363250502946 vs 0.9895957540514192
for
https://www.channelnewsasia.com/news/singapore/ge2020-nomination-tanjong-pagar-grc-pap-psp-lee-hsien-yang-12884760 vs https://www.straitstimes.com/politics/singapore-ge2020-pap-to-face-psp-in-tanjong-pagar-grc
- both on Tanjong pagar GRC, Lee hsien yang, mistakes in form.

Figure A3b: Pairs of self-sourced and self-classified similar news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model



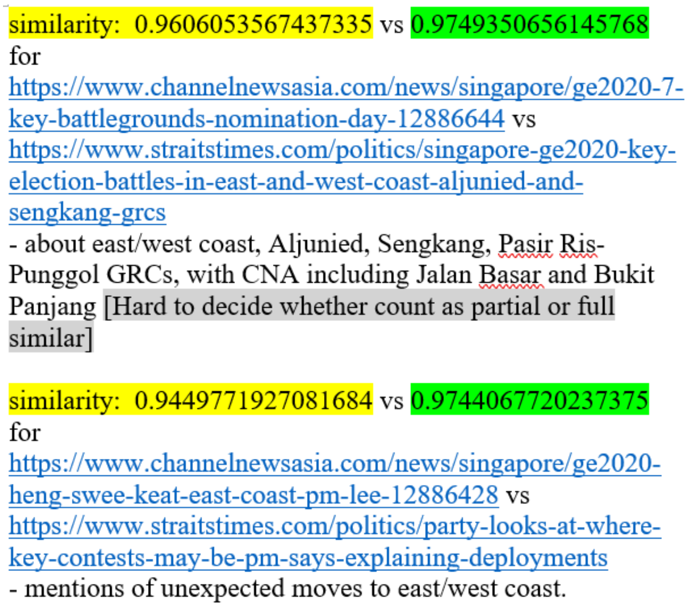
Case 2: Contains some similarity

similarity: 0.9580495639562051 vs 0.9676621888163908
for https://www.straitstimes.com/singapore/singapore-ge-2020-pap-new-face-ivan-lim-withdraws-as-a-candidate vs https://www.channelnewsasia.com/news/singapore/ge2020-pap-jurong-new-candidate-xie-yao-quan-tharman-ivan-lim-12879678
- mentions of Ivan Lim

similarity: 0.9286062675463214 vs (broken link)
for https://www.channelnewsasia.com/news/singapore/covid-19-community-cases-jun-30-citizen-pr-dormitory-worker-12885692 vs https://www.straitstimes.com/singapore/246-new-coronavirus-cases-in-singapore-including-6-in-the-community-78-more-dorms-cleared
- mentions of the 246 new covid cases.

similarity: 0.9709823059873754 vs 0.9793706036064693
for
https://www.channelnewsasia.com/news/singapore/ge2020-nomination-tanjong-pagar-grc-pap-psp-lee-hsien-yang-12884760 vs https://www.straitstimes.com/politics/singapore-ge2020-lee-hsien-yang-not-standing-as-candidate
- mentions of Mr Lee Hsien Yang as a non-candidate

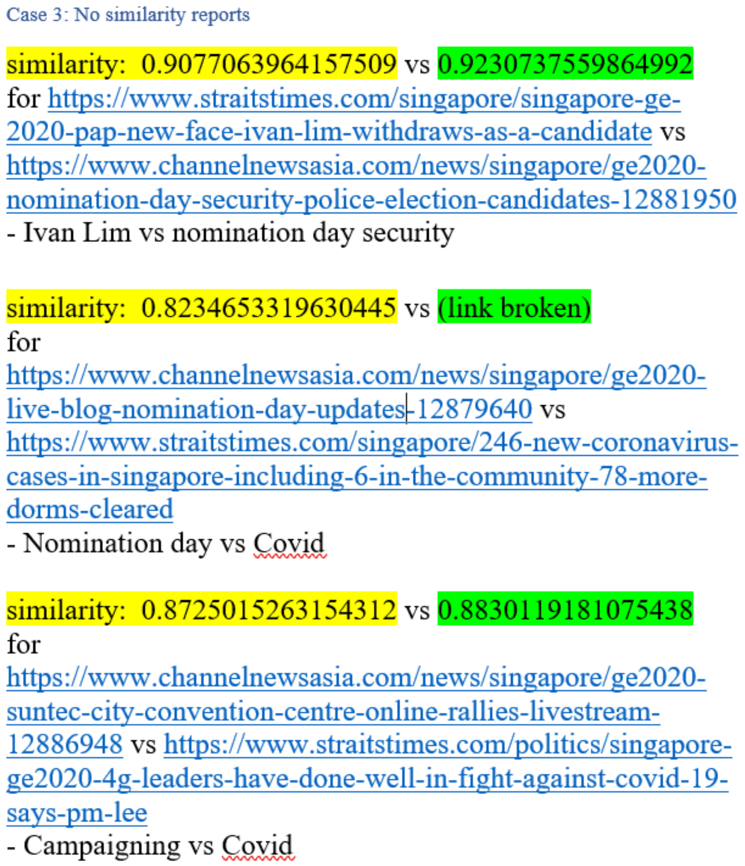
Figure A4a: Pairs of self-sourced and self-classified some similarities news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model



similarity: 0.9606053567437335 vs 0.9749350656145768
for
https://www.channelnewsasia.com/news/singapore/ge2020-7-key-battlegrounds-nomination-day-12886644 vs
https://www.straitstimes.com/politics/singapore-ge2020-key-election-battles-in-east-and-west-coast-aljunied-and-sengkang-grcs
- about east/west coast, Aljunied, Sengkang, Pasir Ris-Punggol GRCs, with CNA including Jalan Basar and Bukit Panjang [Hard to decide whether count as partial or full similar]

similarity: 0.9449771927081684 vs 0.9744067720237375
for
https://www.channelnewsasia.com/news/singapore/ge2020-heng-swee-keat-east-coast-pm-lee-12886428 vs
https://www.straitstimes.com/politics/party-looks-at-where-key-contests-may-be-pm-says-explaining-deployments
- mentions of unexpected moves to east/west coast.

Figure A4b: Pairs of self-sourced and self-classified some similarities news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model



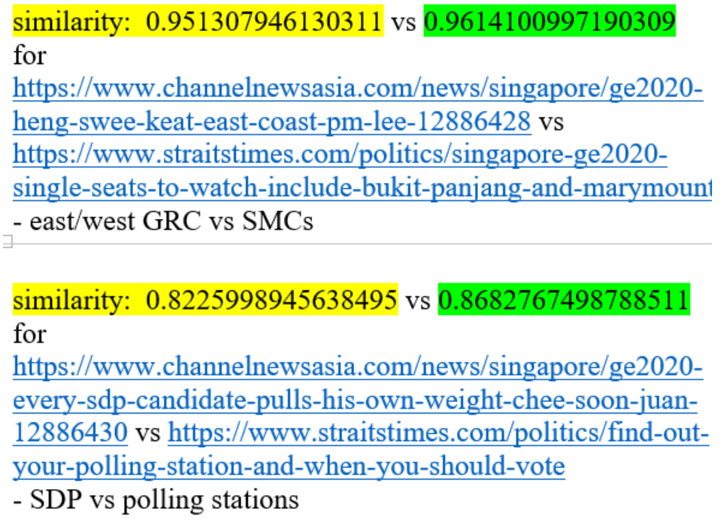
Case 3: No similarity reports

similarity: 0.9077063964157509 vs 0.9230737559864992
for https://www.straitstimes.com/singapore/singapore-ge-2020-pap-new-face-ivan-lim-withdraws-as-a-candidate vs https://www.channelnewsasia.com/news/singapore/ge2020-nomination-day-security-police-election-candidates-12881950
- Ivan Lim vs nomination day security

similarity: 0.8234653319630445 vs (link broken)
for
https://www.channelnewsasia.com/news/singapore/ge2020-live-blog-nomination-day-updates-12879640 vs https://www.straitstimes.com/singapore/246-new-coronavirus-cases-in-singapore-including-6-in-the-community-78-more-dorms-cleared
- Nomination day vs Covid

similarity: 0.8725015263154312 vs 0.8830119181075438
for
https://www.channelnewsasia.com/news/singapore/ge2020-suntec-city-convention-centre-online-rallies-livestream-12886948 vs https://www.straitstimes.com/politics/singapore-ge2020-4g-leaders-have-done-well-in-fight-against-covid-19-says-pm-lee
- Campaigning vs Covid

Figure A5a: Pairs of self-sourced and self-classified not similar news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model

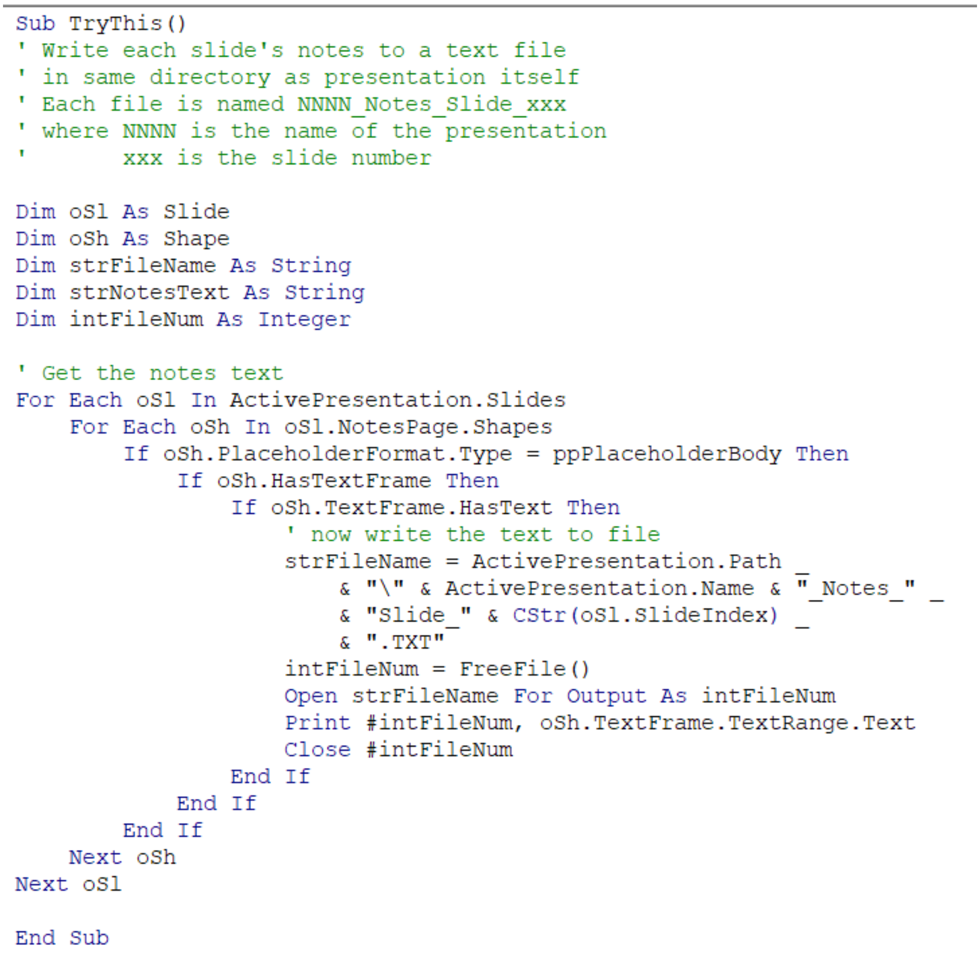


similarity: 0.951307946130311 vs 0.96141009971903 09
for
https://www.channelnewsasia.com/news/singapore/ge2020-heng-swee-keat-east-coast-pm-lee-12886428 vs https://www.straitstimes.com/politics/singapore-ge2020-single-seats-to-watch-include-bukit-panjang-and-marymount
- east/west GRC vs SMCs

similarity: 0.8225998945638495 vs 0.8682767498788511
for
https://www.channelnewsasia.com/news/singapore/ge2020-every-sdp-candidate-pulls-his-own-weight-chee-soon-juan-12886430 vs https://www.straitstimes.com/politics/find-out-your-polling-station-and-when-you-should-vote
- SDP vs polling stations

Figure A5b: Pairs of self-sourced and self-classified not similar news articles from different sources. Yellow highlight is the similarity score output by spaCy default large model, while green highlight is the similarity score output by spaCy - BERT model

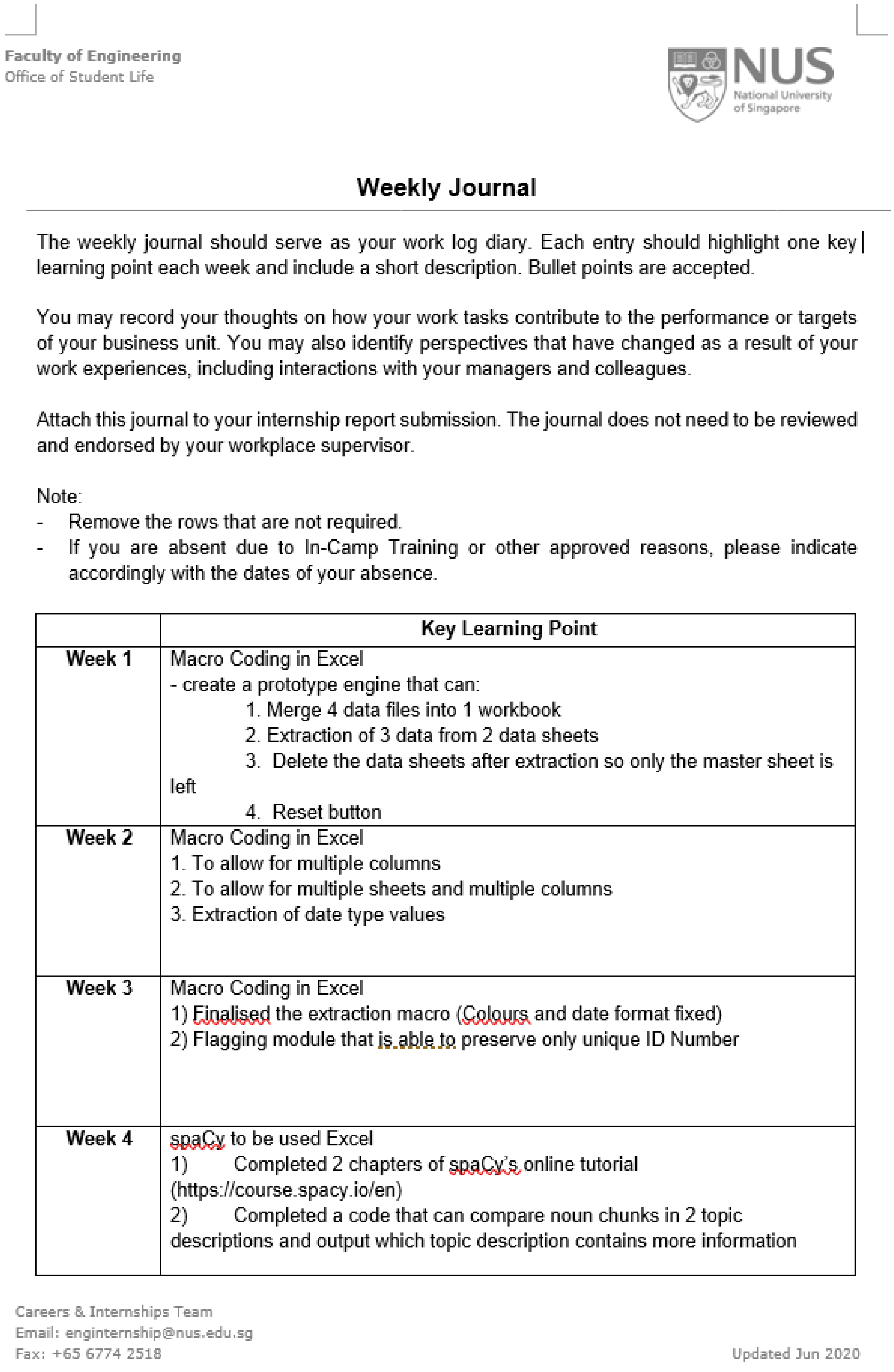
Appendix B : Code used



```
Sub TryThis()
' Write each slide's notes to a text file
' in same directory as presentation itself
' Each file is named NNNN_Notes_Slide_xxx
' where NNNN is the name of the presentation
'       xxx is the slide number

Dim oSl As Slide
Dim oSh As Shape
Dim strFileName As String
Dim strNotesText As String
Dim intFileNum As Integer

' Get the notes text
For Each oSl In ActivePresentation.Slides
    For Each oSh In oSl.NotesPage.Shapes
        If oSh.PlaceholderFormat.Type = ppPlaceholderBody Then
            If oSh.HasTextFrame Then
                If oSh.TextFrame.HasText Then
                    ' now write the text to file
                    strFileName = ActivePresentation.Path _
                        & "\" & ActivePresentation.Name & "_Notes_" _
                        & "Slide_" & CStr(oSl.SlideIndex) _
                        & ".TXT"
                    intFileNum = FreeFile()
                    Open strFileName For Output As intFileNum
                    Print #intFileNum, oSh.TextFrame.TextRange.Text
                    Close #intFileNum
                End If
            End If
        End If
    Next oSh
Next oSl

End Sub
```

Figure B1: Macro Code to be used when generating text files from PowerPoint slides with the script written in the presenter's notes in each slide

Weekly Journal



Faculty of Engineering
Office of Student Life

NUS
National University
of Singapore

Weekly Journal

The weekly journal should serve as your work log diary. Each entry should highlight one key learning point each week and include a short description. Bullet points are accepted.

You may record your thoughts on how your work tasks contribute to the performance or targets of your business unit. You may also identify perspectives that have changed as a result of your work experiences, including interactions with your managers and colleagues.

Attach this journal to your internship report submission. The journal does not need to be reviewed and endorsed by your workplace supervisor.

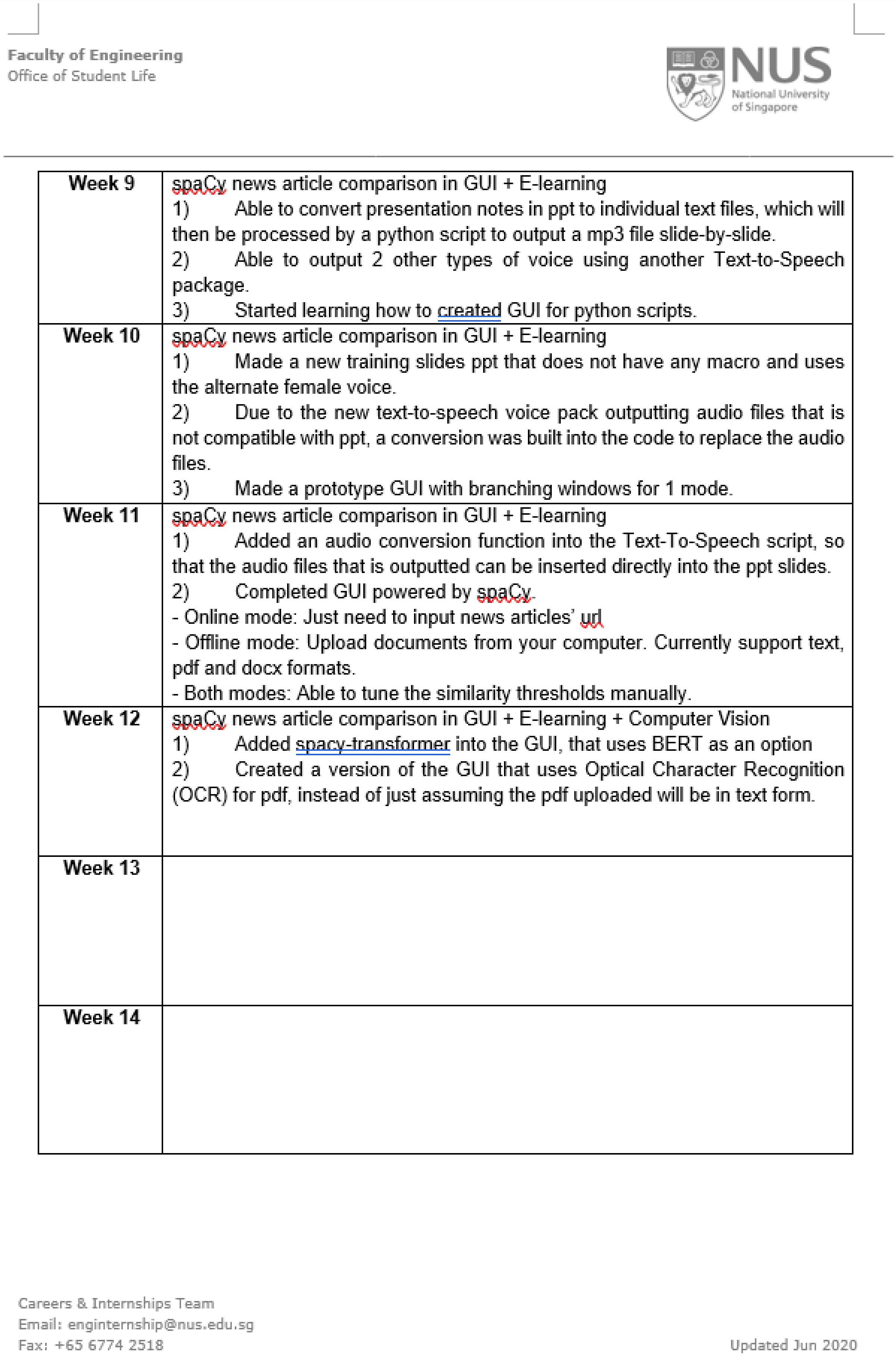
Note:
- Remove the rows that are not required.
- If you are absent due to In-Camp Training or other approved reasons, please indicate accordingly with the dates of your absence.

	Key Learning Point
Week 1	Macro Coding in Excel - create a prototype engine that can: 1. Merge 4 data files into 1 workbook 2. Extraction of 3 data from 2 data sheets 3. Delete the data sheets after extraction so only the master sheet is left 4. Reset button
Week 2	Macro Coding in Excel 1. To allow for multiple columns 2. To allow for multiple sheets and multiple columns 3. Extraction of date type values
Week 3	Macro Coding in Excel 1) Finalised the extraction macro (Colours and date format fixed) 2) Flagging module that is able to preserve only unique ID Number
Week 4	spaCy to be used Excel 1) Completed 2 chapters of spaCy's online tutorial (https://course.spacy.io/en) 2) Completed a code that can compare noun chunks in 2 topic descriptions and output which topic description contains more information

Careers & Internships Team
Email: enginternship@nus.edu.sg
Fax: +65 6774 2518

Updated Jun 2020

	3) Completed a code that can compare how similar to strings are Observation: The built-in similarity function in spaCy is very inaccurate according to my testing with strings given by Alvin and YY's team. Reading through the link given by Professor Ng to see which one best suit this context.
Week 5	spaCy to be used Excel 1) Combination.py is the code I made by combining both codes from last week. This allows me to compare 2 text's nouns to see how similar they are. 2) Combination_V2.py shares similar core functions to Combination.py but both text first undergoes a process that removes stop words and punctuations, lemmatise each word and convert all characters to lowercase. After that, if the similarity score is low (currently set to <= 0.6), it would ask for user input on whether both text are similar.
Week 6	spaCy to be used Excel 1) Created Suggestion.py that suggest to user which text to use base on similarity index and how "rich" the text is. 2) Manually extracted 10 test cases base on the 2 scenarios that spaCy can be used each. From there, test results were analyzed to choose which model to use moving forward and what is the cut off value between Similar and Not Similar.
Week 7	spaCy + Excel testing 1) Able to excess python script from Excel VBA. 2) Able to export values from python script to an Excel cell. 3) Able to import text from 2 cells in an Excel sheet to spaCy python script, run the script to give us similarity score and export the "richer" text to an Excel cell.
Week 8	spaCy news article comparison + E-learning 1) Made a python script that accepts 2 news article url and outputs a similarity score. 2) Tested the script with 3 test cases (Same reports, reports with similarity, different reports). Each test case has 5 comparisons as seen in my previous email. [After a meeting with Prof Ng, I might have to rearrange some of the comparisons] 3) Created a simple Text-to-Speech script that doesn't need internet access. However, changing the package to Google Text-to-Speech suggested by Prof Ng.



Week 9	spaCy news article comparison in GUI + E-learning 1) Able to convert presentation notes in ppt to individual text files, which will then be processed by a python script to output a mp3 file slide-by-slide. 2) Able to output 2 other types of voice using another Text-to-Speech package. 3) Started learning how to created GUI for python scripts.
Week 10	spaCy news article comparison in GUI + E-learning 1) Made a new training slides ppt that does not have any macro and uses the alternate female voice. 2) Due to the new text-to-speech voice pack outputting audio files that is not compatible with ppt, a conversion was built into the code to replace the audio files. 3) Made a prototype GUI with branching windows for 1 mode.
Week 11	spaCy news article comparison in GUI + E-learning 1) Added an audio conversion function into the Text-To-Speech script, so that the audio files that is outputted can be inserted directly into the ppt slides. 2) Completed GUI powered by spaCy. - Online mode: Just need to input news articles' url - Offline mode: Upload documents from your computer. Currently support text, pdf and docx formats. - Both modes: Able to tune the similarity thresholds manually.
Week 12	spaCy news article comparison in GUI + E-learning + Computer Vision 1) Added spacy-transformer into the GUI, that uses BERT as an option 2) Created a version of the GUI that uses Optical Character Recognition (OCR) for pdf, instead of just assuming the pdf uploaded will be in text form.
Week 13	
Week 14	