INVESTIGATION REPORT

Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques

By

HERMANTO

TP054802

UC3F2111CS(DA)

A report submitted in partial fulfillment of the requirements of Asia Pacific University of Technology and Innovation for the degree of

B.Sc. (Hons) Computer Science (Data Analyst)

Supervised by Dr. Dewi Octaviani

2nd Marker: Ms. Minnu Hellen Joseph

Mar-2

Table of Content

CHAPTER 1: INTRODUCTION TO THE STUDY **1**
1.1. Background to the Project 1
1.2. Problem Statements 1
1.3. Rationale 2
1.4. Potential Benefits 3
1.4.1 Tangible Benefits 3
1.4.2 Intangible Benefits 3
1.5 Target Users 4
1.6 Scope and Objectives 4
1.6.1 Aim 4
1.6.2 Objectives 4
1.6.3 Deliverables 5
1.6.4 Nature of Challenges 5
1.7 Overview of Report 5
1.8 Project Plan 6

CHAPTER 2: LITERATURE REVIEW **10**
2.1 Introduction 10
2.2 Domain Research 10
2.2.1 E-Commerce 10
2.2.2 Prediction Model 11
2.2.3 Machine Learning for E-Commerce 16
2.2.4 Text Mining of Unstructured Textual Data 16
2.3 Similar Models 17
2.4 Summary 18

CHAPTER 3: TECHNICAL RESEARCH **19**
3.1 Programming Language Chosen 19
3.1.1 R Language 19
3.1.2 Phyton 19
3.1.3 Summary 20
3.2 IDE Chosen 20
3.2.1 Machine Learning Development (Jupyter Notebook) 20
3.2.2 Website Development (PyCharm) 21
3.3 Libraries chosen/ Tools chosen 21
3.3.1 pandas 21
3.3.2 string 21

3.3.3 tqdm 22
3.3.4 random 22
3.3.5 json 22
3.3.6 os 22
3.3.7 ast 23
3.3.8 spaCy 23
3.3.9 scikit-learn 23
3.3.10 Flask 23
3.3.11 matplotlib 23
3.3.12 SciPy 24
3.3.13 Seaborn 24
3.4 Operating System Chosen 25
3.4.1 Hardware requirement 25
3.4.2 Software requirement 25
3.5 Web server chosen 26
3.6 Web browser chosen 26
3.7 Summary 26

CHAPTER 4: METHODOLOGY **27**
4.1 Introduction 27
4.1.1 Comparison Table of SEMMA, CRISP-DM & KDD 27
4.1.1.1 CRISP-DM 27
4.1.1.2 SEMMA 27
4.1.1.3 KDD 28
4.1.2 Summary 28
4.2 Methods 28
4.2.1 Business Understanding 30
4.2.2 Data Understanding 30
4.2.2.1 Basic Text Data-Preprocessing and Data Cleaning 31
4.2.2.2 Data Exploration for Text Data 31
4.2.3 Data Preparation 33
4.2.3.1 Tokenization and Labelling for Named Entity Recognition Task 34
4.2.4 Modeling 35
4.2.4.1 Selecting Modeling Techniques 35
4.2.4.2 Generate Test Design 35
4.2.4.3 Assessing the Model 35
4.2.4.4 Building the Model 36
4.2.4.5 Spelling Correction Model 38
4.2.5 Evaluation 39

4.2.6 Deployment 39
4.3 Summary 40

CHAPTER 5: DATA ANALYSIS **41**
5.1 Introduction 41
5.2 Data Collection 41
5.3 Data Understanding/Exploration 44
5.3.1 Data Preparation 44
5.3.2 Basic Data Exploration 45
5.3.3 Semantic Analysis 47
5.3.4 Text Length Analysis 48
5.4 Data Visualization 53
5.4.1 Top Unigram Distribution 53
5.4.2 Top Bigram Distribution 55
5.4.3 Top Trigram Distribution 58
5.4.4 Most Common Words 60
5.4.5 Word Cloud 61
5.5 Data Preprocessing 62
5.5.1 Dataset Splitting 62
5.5.2 Data Cleaning 63
5.5.3 Tokenization and Labelling 63
5.5.4 Building Word List & Token Labelling 64
5.6 Model Buildings 66
5.6.1 Introduction 66
5.6.2 Tokenization 67
5.6.3 Processor 68
5.6.4 Databunch 69
5.6.5 Training 69
5.7 Summary 70

CHAPTER 6: RESULTS AND DISCUSSION **71**
6.1 Introduction 71
6.2 Model Evaluation 71
6.3 Model Deployment 74
6.3.1 Website Design 74
6.3.2 Deployment with Flask 76
6.4 Results and Discussion 77

CHAPTER 7: CONCLUSIONS AND REFLECTIONS **78**
7.1 Conclusions 78

7.2 Reflections 79
7.3 Future Works 79

REFERENCES
APPENDICES
FAST TRACK FORM
DISCLAIMER ETHICS FORM
GANTT CHART
PPF
PSF

Table of Figures

CHAPTER 1: INTRODUCTION TO THE STUDY **1**
Figure 1.1: Process of Geocoding 2

CHAPTER 2: LITERATURE REVIEW **9**
Figure 2.1: RNN Model Description 10
Figure 2.2: BERT Model Example 11
Figure 2.3: CRF Graphical Model 13

CHAPTER 4: METHODOLOGY **22**
Figure 4.1: KPI of Shopee in Indonesia 27
Figure 4.2: Sample of the Train dataset 27
Figure 4.3: Missing and Duplicate Values Percentage 28
Figure 4.4: Text Length Distribution 29
Figure 4.5: Semantic Analysis 29
Figure 4.6: Overview of Name Entity Recognition Model 33

List of Tables

CHAPTER 1: INTRODUCTION TO THE STUDY **1**
Table 1.1: Project Plan Table 6

CHAPTER 2: LITERATURE REVIEW **9**
Table 2.1: Similar Models Table 15

CHAPTER 3: TECHNICAL RESEARCH **17**
Table 3.1: Hardware Requirement 21
Table 3.2: Software Requirement 21

CHAPTER 4: METHODOLOGY **22**
Table 4.1: Summarized Strengths and Disadvantages 24
Table 4.2: Sample Data 30
Table 4.3: Sample data when included labelling 34

Acknowledgement

This project would not have been possible without the support of many people. Many thanks to my adviser, Dr Dewi Octaviani, who read my numerous revisions and helped make some sense of the confusion. With her continuous offered guidance and support. Besides my advisor, I would like to thank the rest of my thesis committee: Mr Dhason Padmakumar for clearing my doubts and provide insightful comments as well as answering my questions.

I am deeply indebted to my respected lecturers and other members of Computing and IT department who patiently imparted their knowledge for us to acquire a better future.

I also would like to thank my fellow classmates for the attention and help to clear my doubts when I have certain questions. As well as during the assignment where we were working together before deadlines. Thank you for the three years of fun.

Thanks to Asia Pacific University for providing me guidance to complete this project and made me who I am today.

Thanks to my parents for providing me their support, love and endure with me at home. Thank you for providing me the support emotionally and financially

CHAPTER 1: INTRODUCTION TO THE STUDY

1.1 Background to the project

Throughout our daily life, people ought to buy something whether it is grocery or daily necessity to fulfill daily needs. Thus, surrounding the environment there will always be at least a grocery store that sells daily necessities. But as time progresses people find it a hassle to buy items at the grocery store or market. It is a hassle to drive their way to the store even though it's nearby, and you will always need to consider time, weather conditions, parking car, lining-up, and so on. There are too many disadvantages which are why there exists online retail marketing. Online retail service is one of the most popular topics as they are rapidly expanding due to people being prevented from going out of their house due to COVID-19. According to OECD (2020) the sales of online retail by 30% in April 2020 compared to April 2019. The online retail service allows them to have their desired items to be delivered right in front of their house, at the desired time and date without much hassle at all. As a result, many people are willing to use online retail services. Additionally, the current trend of E-commerce in 2022 from Hung (2022) shows that globally e-commerce will account 20.4% of global retail sales. Its rapid growth are tremendous causing people to focus on the E-commerce.

Recently, from the perspective of the retail marketing service in Indonesia, the marketer encountered a problem. The addresses entered by the customer are often not clear or incomplete thus the system cannot recognize the address when it passes for geocoding. Therefore, the aim of the project is to apply Data Mining Technique to help the extraction of the address entered by customers, predicting the point of address and street. and completing the incomplete words.

1.2 Problem statements

The address element extraction purpose is to ensure the address elements that were passed for geocoding can be geocoded. According to Chakravarty (2018), geocoding allows address standardization which makes it easier for finding the exact address and coordination that can be pointed on the map. Thus, using geocoding can be useful to perform analysis of a region or market that can be used for drawing insightful information and making a decision.

Following figure 1, the process of geocoding simply needed the standardized version of address however that is not easily achieved as most of the input entered by customers are either incomplete or unstructured thus there's a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it's common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. Therefore, a data analyst is assigned to solve this problem, correcting, and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding.

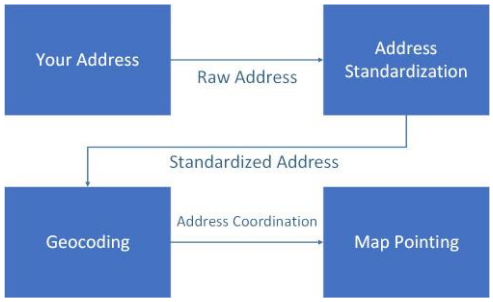


Figure 1.1: Process of Geocoding

1.3 Rationale

Customers tend to enter incomplete or unstructured, thus there is a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it is common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. Therefore, data analyst is assigned to solve this problem, correcting, and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding

1.4 Potential benefits

This research will benefit greatly for the online retail sectors and marketing, accurately predicting the element of incomplete address that will play an important role in the implementation of geocoding to obtain the geographic coordinates to deliver the parcel, ensuring efficient and high-quality customer satisfaction with the quick arrival of the parcel. The common occurrence for customers mistyping or entering incomplete addresses encourages the demand for the proposed system and for the online retail company that is planning for the implementation of geocoding. The inventory department that oversees finding parcel addresses will utilize the system, saving human resources and the need to manually find the accurate address of parcels.

1.4. 1 Tangible benefits

1. Lower the costs of labor and time with the improvement of efficiency as the machine will do the task automatically
2. Allow the company to analyze the address data as the address is already standardized into required formats
3. Allow the company to identify the incorrect address and make the decision from there

1.4. 2 Intangible benefits

1. It increases customer experience as there won't be any incorrect address being sent to the system, thus parcel or item will arrive at the correct addresses.
2. It gives much more accurate results compared to being done by an employee and can reduce the labor
3. Lower the risk of sending the parcel to the incorrect addresses

1.5 Target users

E-commerce will require customers to enter or type in their addresses when buying products from the website. The address that was received will then be reviewed by someone to check the location. The standardization process will be completed following the standards. Most importantly, the model will be unsupervised, which means the task is conducted automatically and have stable accuracy.

1.6 Scope and objectives

1.6.1 Aim

Customers tend to enter incomplete or unstructured, thus there is a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it is common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. This is why a data analyst is assigned to solve this problem, correcting and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding

1.6.2 Objectives

i. To determine suitable techniques for extracting unstructured e-commerce data.

ii. To implement data mining techniques for unstructured addresses for extracting a point of interest and street address.

iii. To implement data mining techniques to develop a spelling correction model and standardization of the address to complete and allow geocoding analysis.

iv. To evaluate the spelling correction model and the geocoding analysis result.

1.6. 3 Deliverables - Functionality of the proposed system

1. To collect a raw dataset of addresses from an e-commerce website
2. To perform business understanding the impact of the model, determine business objectives, and assess the situation to determine the goal
3. To conduct data pre-processing of the raw address data suitable to be tokenization and BERT-readable
4. To train and develop the spelling correction model to identify mistakes customers made
5. To train and develop element extraction model to extract point of interest and street address from the raw data
6. To evaluate the NER and spelling correction model using the determined accuracy scoring metrics.

1.6.4 Nature of Challenges

The challenge would be developing two main models for the machine learning algorithm. The first challenge would be developing the Named Entity Recognition Model to recognize the tokenized words and to correctly predict the specific elements. Following Figure 5, the visualization of Name Entity Recognition can be seen as it utilizes the BERT model to predict the range of tokens that are the answer, meaning it was framed as the QA Problem Model.

The second part of the model would be developing a spelling correction model which is necessary to increase the accuracy of prediction as the data often has incomplete words in both elements. Thus, the model is deployed to correct the labeled token which is not complete.

1.7 Overview of this report

Chapter 1 will provide the overall description of the report, including the objectives and aim of the project, the benefits, and the challenges. In chapter 2, a deep literature review is written to conduct research and gain knowledge of the project. The knowledge domain research includes Text mining

of unstructured data, E-commerce, Prediction model, and machine learning for e-commerce. For chapter 3, it discusses the technical part of the project which includes the computer language that will be used to build the machine learning model, the IDE, and the basic requirement to run the IDE or build the model. For chapter 4, the methodology is discussed in detail, first, there will be a comparison between two models which are CRISP-DM and SEMMA, then the decision will be made between the two which one is chosen for the project. Afterward, in detail describe the methodology of CRISP-DM and how will it be applied in the project. Examples are also provided in tables and figures to show how it was done or to give an overall expectation. Lastly, chapter 5 will sum up the whole report and contain reflections of the author.

1.8 Project Plan

Table 1.1: Project Plan Table

Task ID	Task Name	Duration (Day)	Start Date	End Date	Status
1	**Chapter 1: Introduction to the Study**	**6**	**12/6/21**	**Fri 12/11/21**	**Done**
1.1	Background of the project	1	12/6/21	12/6/21	Done
1.2	Problem Statements	1	12/7/21	12/7/21	Done
1.3	Rationale	1	12/8/21	12/8/21	Done
1.4	Potential Benefits	1	12/8/21	12/8/21	Done
1.4.1	Tangible Benefits	1	12/9/21	12/9/21	Done
1.4.2	Intangible Benefits	1	12/9/21	12/9/21	Done
1.5	Target users	1	12/10/21	12/10/21	Done
1.6	Scope and objectives	1	12/10/21	12/10/21	Done
1.6.1	Aim	1	12/10/21	12/10/21	Done
1.6.2	Objectives	1	12/10/21	12/10/21	Done
1.6.3	Deliverables	1	12/10/21	12/10/21	Done
1.6.4	Nature of Challenges	1	12/10/21	12/10/21	Done
1.7	Overview of Report	1	12/11/21	12/11/21	Done

1.8	Project Plan	1	12/11/21	12/11/21	Done
2	**Chapter 2: Literature Review**	**5**	**12/12/21**	**12/16/21**	**Done**
2.1	Introduction	1	12/12/21	12/12/21	Done
2.2	Domain Research	2	12/13/21	12/14/21	Done
2.2.1	E-Commerce	1	12/13/21	12/13/21	Done
2.2.2	Prediction Model	1	12/13/21	12/13/21	Done
2.2.3	Machine Learning for E-Commerce	1	12/14/21	12/14/21	Done
2.2.4	Text Mining of Unstructured Data	1	12/14/21	12/14/21	Done
2.3	Similar Models	1	12/15/21	12/15/21	Done
2.4	Summary	1	12/16/21	12/16/21	Done
3	**Chapter 3: Technical Research**	**5**	**12/17/21**	**12/21/21**	**Done**
3.1	Language chosen	**2**	12/17/21	12/17/21	Done
3.1.1	R Language	1	12/17/21	12/17/21	Done
3.1.2	Phyton	1	12/17/21	12/17/21	Done
3.1.3	Summary	1	12/17/21	12/17/21	Done
3.2	IDE (Interactive Development Environment) chosen	1	12/18/21	12/18/21	Done
3.3	Libraries chosen/ Tools chosen	1	12/19/21	12/19/21	Done
3.3.1	pandas	1	12/19/21	12/19/21	Done
3.3.2	string	**1**	12/19/21	12/19/21	Done
3.3.3	tqdm	**1**	12/19/21	12/19/21	Done
3.3.4	random	**1**	12/19/21	12/19/21	Done
3.3.5	json	**1**	12/19/21	12/19/21	Done
3.3.6	os	**1**	12/19/21	12/19/21	Done
3.3.7	ast	**1**	12/19/21	12/19/21	Done
3.4	Operating System Chosen	**1**	12/20/21	12/20/21	Done
3.4.1	Hardware requirement	**1**	12/19/21	12/19/21	Done
3.4.2	Software requirement	**1**	12/19/21	12/19/21	Done

3.5	Web server chosen	**1**	12/20/21	12/20/21	Done
3.6	Web browser chosen	**1**	12/20/21	12/20/21	Done
3.7	Summary	**1**	12/21/21	12/21/21	Done
4	**Chapter 4: Methodology**	**10**	**12/22/21**	**12/31/21**	**Done**
4.1	Introduction	1	12/22/21	12/22/21	Done
4.1.1	Comparison Table of SEMMA and CRISP-DM	1	12/22/21	12/22/21	Done
4.1.2	Summary	1	12/22/21	12/22/21	Done
4.2	Methods	7	12/23/21	12/25/21	Done
4.2.1	Business Understanding	1	12/23/21	12/23/21	Done
4.2.2	Data Understanding	1	12/24/21	12/24/21	Done
4.2.3	Data Preparation	1	12/25/21	12/25/21	Done
4.2.4	Modeling	4	12/26/21	12/29/21	Done
4.2.4.1	Selecting Modeling Techniques	1	12/26/21	12/26/21	Done
4.2.4.2	Generate Test Design	1	12/26/21	12/26/21	Done
4.2.4.3	Assessing The Model	1	12/27/21	12/27/21	Done
4.2.4.4	Building The Model	1	12/28/21	12/28/21	Done
4.2.4.5	Spelling Correction Model	1	12/29/21	12/29/21	Done
4.2.5	Evaluation	2	12/30/21	12/30/21	Done
4.2.6	Deployment	1	12/31/21	12/31/21	Done
4.3	Summary	1	12/31/21	12/31/21	Done
5	**Chapter 5: Data Analysis**	**10**	**6/4/22**	**6/14/22**	**Done**
5.1	Introduction	1	6/4/22	6/5/22	Done
5.2	Data Collection	1	6/4/22	6/5/22	Done
5.3	Data Understanding/Exploration	2	6/4/22	6/6/22	Done
5.3.1	Data Preparation	1	6/4/22	6/5/22	Done
5.3.2	Basic Data Exploration	1	6/4/22	6/5/22	Done
5.3.3	Semantic Analysis	1	6/5/22	6/6/22	Done
5.3.4	Text Length Analysis	1	6/5/22	6/6/22	Done
5.4	Data Visualization	2	6/6/22	6/8/22	Done

5.4.1	Top Unigram Distribution	1	6/6/22	6/7/22	Done
5.4.2	Top Bigram Distribution	1	6/6/22	6/7/22	Done
5.4.3	Top Trigram Distribution	1	6/6/22	6/7/22	Done
5.4.4	Most Common Words	1	6/7/22	6/8/22	Done
5.4.5	Word Cloud	1	6/7/22	6/8/22	Done
5.5	Data Preprocessing	2	6/8/22	6/10/22	Done
5.5.1	Dataset Splitting	1	6/8/22	6/9/22	Done
5.5.2	Data Cleaning	1	6/8/22	6/9/22	Done
5.5.3	Tokenization and Labelling	1	6/9/22	6/10/22	Done
5.5.4	Building Word List & Token Labelling	1	6/9/22	6/10/22	Done
5.6	Model Buildings	4	6/10/22	6/14/22	Done
5.6.1	Introduction	1	6/10/22	6/11/22	Done
5.6.2	Tokenization	2	6/10/22	6/12/22	Done
5.6.3	Processor	1	6/12/22	6/13/22	Done
5.6.4	Databunch	1	6/12/22	6/13/22	Done
5.6.5	Training	1	6/12/22	6/13/22	Done
5.7	Summary	1	6/13/22	6/14/22	Done
6	**Chapter 6: Results and Discussion**	**15**	**6/15/22**	**6/30/22**	**Done**
6.1	Introduction	1	6/15/22	6/16/22	Done
6.2	Model Evaluation	5	6/16/22	6/21/22	Done
6.3	Model Deployment	7	6/21/22	6/28/22	Done
6.3.1	Website Design	5	6/21/22	6/26/22	Done
6.3.2	Deployment with Flask	2	5/26/22	6/28/22	Done
6.4	Results and Discussion	1	6/29/22	6/30/22	Done
7	**Chapter 7: Conclusions and Reflection**	**3**	**1/1/22**	**1/1/22**	**Done**
7.1	Conclusions	1	1/2/22	1/2/22	Done
7.2	Reflections	2	1/3/22	1/4/22	Done

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The continuously evolving of the Internet, also causes the E-commerce industry to evolve, adapting to the strict environment that is required for online shopping. For businessmen of E-commerce assuring the customer receive the product they ordered is the number one priority to assure customer experience. Thus, identifying the knowledge required to build a machine-learning algorithm to accurately predict customer address is essential as it enables the company to not waste time and resources as well as improve the overall operational performance.

2.2 Domain Research

2.2.1 E-Commerce

E-commerce has always been constantly moving, especially during the pandemic the industry of online retail improved tremendously (Alfonso, et al., 2021). Their research shows the statistic of E-commerce sales volumes since the pandemic, comparing the statistics between countries. The three main factors that needed to be considered during these times are product as during the pandemic few companies must stop producing product thus the availability of the product is a concern. Second, e-commerce has been subject to disruptions withinside the supply of services to resource distribution, shipment, and after-sale requirements with the resource of the use of customers. Third, the extended shift to digital purchases moreover highlighted online patron protection as one of the vital challenges. Reports of fraudulent and dishonest practices progressed sooner or later with the lockdowns. According to Taylor's (2019) prediction, the market of E-commerce will increase by up to 25% by the year 2026. There’s also a study of Escursell and his team (2021), who researched sustainability in E-commerce packaging, there’s data extraction involved where it analyses the percentage of transport packaging and primary packaging and another packaging in terms of energy needed.

2.2.2 Prediction Model

RNN Model

The study of Li and his team (2020) predicted the English version from the Chinese address. It uses a recurrent neural network (RNN) for address segmentation. The method of segmentation is first by inputting the Chinese address sequence which will then be vectorized through the neural network. After that, the Viterbi algorithm is used for the last segmentation and tagging. It is one of the most powerful algorithms as it's the only algorithm that has an internal memory making it powerful and robust (Donges, 2019). The way RNN works is like the feed-forward neural network and sequential data.

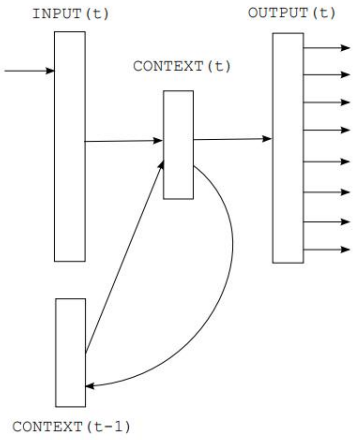


Figure 2.1: RNN Model Description
(Mikolove, Karafiat, Burget, Cernocky, & Khudanpur, 2010)

Based on figure 2.1, the recurrent network model has input, context, and output layer. If input layer as x, context layer as s, output layer as y and current word as ω. The input vector of x (t) is calculated by concatenating vector ω and output of s at time t – 1 where they are usually merged into one token with the following model descriptions equation:

$$x(t) = \omega(t) + s(t+1)$$

$$s_j(t) = f\left(\sum_i x_i\,(t) u_{ji}\right)$$

$$y_{k(t)} = g\left(\sum_j s_j(t) v_{kj}\right)$$

Where $f(z)$ is sigmoid activation function with the following equation:

$$f(z) = \frac{1}{1+e^{-z}}$$

And $g_{(z)}$ is softmax function with the following equation:

$$g_{(z)} = \frac{e^{z_m}}{\Sigma_k e^{z_k}}$$

BERT Model

Another research of information extraction made by Gupta and Nishu (2020), their method of extraction is language-based, done by the Bidirectional Encoder Representations from Transformers (BERT) model where it assigns tokens to each word which later be used for the Named Entity Recognition process. The BERT model is essentially a model published by researchers of Google's AI language (Horev, 2018). The model makes use of the "transformer" which learns the relation between words in a text.

The BERT Architecture usually consists of two variants. The BERT base consists of 12 layers, 12 attention heads. and 110 million parameters and the other variant is called BERT Large which consists of twice the number of layers, 16 attention heads, and 340 million parameters.

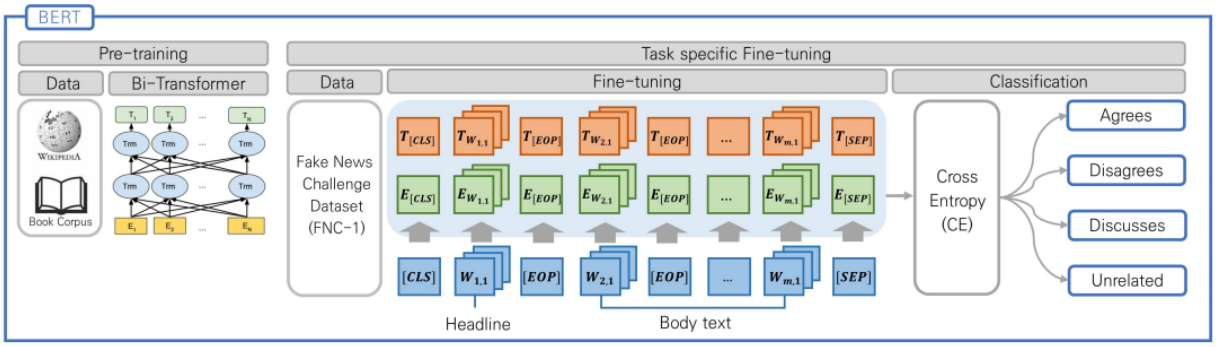


Figure 2.2: BERT Model Example (Li, Jin, Liu, Rawat, Cai, & Yu, 2019)

From figure 2.2, the proposed model consisted of three types of embeddings, position embedding, segment embedding, and token embedding. These embeddings are to capture sequences of information and make segments for example taking pairs of sentences to be put as input.

CRF Model

Information extraction is done by Zhang and his team (2019), extracting keywords automatically using the Conditional Random Fields (CRF) model, using two approaches. One of the approaches is keyword extraction where the words are analyzed to identify the words that are relational with each other based on their frequency rate and word lengths. The second approaches are keyword assignments where the keywords are specifically chosen from their vocabulary terms and are classified according to the content which will be related to the vocabulary terms of the keywords. Based on Wallach's (2004) Conditional Random Fields, the notation was simplified into

$$s(y_1, x, i) = s(y_i - 1, y_i, x, i)$$

and

$$F_j(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \boldsymbol{x}, i),$$

where each $f_j\{y_{i-1}, x, i)$ is either state function or transition function. The following are similar equation used by Zheng and his team (2015) when building the CRF model for his project.

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{M} w^{(m)} k_G^{(m)}(\mathbf{f}_i, \mathbf{f}_j),$$

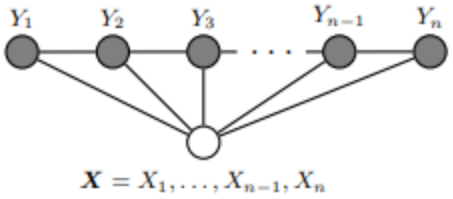


Figure 2.3: CRF Graphical Model

(Wallach, 2004)

N-Gram Model

Song and Croft (2015) define N-Gram Model as a general statistical language that is usually used for retrieving information. Most of the time the statistical language is used to predict the key elements and sequences of words by relying on a large corpus of documents. According to research (Eugene, 1994; Song and Croft, 2015; Ahmed, William De Luca, and Nürnberger, 2009), The statistical language model has been used for recognizing the voice of different people and parsing sentences synthetically. Their research also did comparisons between other language models, and it was statistically proved that the statistical model the team developed is better than Ponte and Croft's language model. The following are equations for *sequential maximum likelihood estimation* (Brown, Della, Desouza, Lai, & Mercer, 1992):

$$\Pr\left(w_n \mid w_1^{n-1}\right) = \frac{C(w_1^{n-1} w_n)}{\sum_w C(w_1^{n-1} w)}.$$

According to Brown and his team, the model is not a consistent model for small values and is more suitable for a large amount of data. They conducted research with 365 million English words, and they have concluded that as n increases, the accuracy of the model also increases but it depends on the parameter estimation that was drawn.

CNN Model

CNN is a deep learning model for processing data with a grid pattern. CNN was inspired by the structure of animal visual cortex. It was created to learn automatically and adaptively spatial per level from the highest to lowest (Yamashita, Nishio, Do & Togashi, 2018). There are several

layers, convolution, pooling, and fully connected. The mentioned layers use mathematical construction known as CNN. The use of Convolution and pooling layers are feature extractions. There's also fully connected layer that can do classification. The model is made up of mathematical operation that include convolution which is crucial to be included in the model as it has linear operation.

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

The formula is for convolution layer, it applies if there's input of W * W * D and D out is number of kernels. F is spatial size, S is the stride, and P is the amount of padding. By calculating an aggregate statistic from the surrounding outputs, the pooling layer substitutes for the network's output at specific locations. This aids in shrinking the representation's spatial size, which lowers the amount of computation and weights needed. Each slice of the representation is subjected to the pooling operation separately.

$$W_{out} = \frac{W - F}{S} + 1$$

The formula is for pooling layer where it applies if the activation map is the size of W * W * D, where F is the spatial size of the kernel, and stride is S. This allows the pooling layer to provide the invariance value that are recognizable (Mishra, 2020).

CFG Model

Context-free languages are described using context-free grammars which is a set of recursive rules that are used to create pattern of the sentence. The study of context-free grammars focusses on languages, compiler design, and theoretical computer science. CFGs are used to describe computer languages, and context-free grammars can be used to automatically produce compiler parser programs. A context-free grammar can be described by a four-element (*V*, Σ, *R*, *S*) where (Moore, Chumbley & Khim, 2022):

- *V* is a finite set of variables
- Σ is a finite set of terminal symbols

- *R* is a set of production rules where each production rule maps a variable to a string
- *S* which is a start symbol.

2.2.3 Machine Learning for E-Commerce

Various fields use machine learning as it is fast, accurate, and highly technical. According to Jean and his team (2016), any domestic and foreign experts' machine learning algorithm could be used to improve economic condition if it is used correctly. The team also managed to run a project for estimation using the inexpensive but accurate model. Liu and his team (2020) examine how machine learning can be used in the E-commerce industry to build the platform for customer repurchase prediction models. The author discusses the various model and one of the models is the XGBoost algorithm to build the prediction model which is based on a customer behavioral system. In the case of Fazal-e-Hasan and his team (2018), take consideration of customers' opinion, meaning they consider customer expectation and expects the customers to engage with them to provide insights of what market brand they are interested to see to meet their satisfaction. Xu and his team (2021) dedicated their work to imparting complicated theoretical know-how of product embedding for e-trade device learning, also set up the equivalence among education product embedding and enough measurement discount with admire to the product relatedness measure.

2.2.4 Text Mining of Unstructured Textual Data

Thavavel and Sivakumar (2012) proposed and designed new generalized body paintings for privateness maintenance in allotted facts mining for unstructured facts surroundings and enforcing the trying out of similarity amongst unstructured textual content. Miller (2002) discusses an unstructured data environment, in which the system is designed to facilitate the interplay of dependent and unstructured facts. The primary functions of the view mechanism, especially as they relate to textual files are supplied withinside the paper. It additionally seems at how the perspectives technique permits the interplay among the facts taken from dependent semi-dependent and unstructured facts sources. Rajman and his team (1998) researched text mining specifically extracting data out of unstructured textual data, where they show off two examples of

extraction using two different text mining models, one of them is automated keyword association extraction and the other is prototypical document mining.

2.3 Similar Models

Table 2.1: Similar Models Table

Author	Features	Text Mining Technique
Antons, Grunwald, Cichy, & Salge (2020)	Applying Text Mining in the field of innovation research	Classification and Clustering
Lee, Yoon, Kim, Kim, So, & Kang (2020)	Extracting valuable information from biomedical literature	BERT Model
Choi, Park, & Kim (2019)	Analysis of BTS fever	Text Term Frequency
Zhang, Fleyeh, Wang, & Lu (2019)	Analysis of construction site accident	SVM, KNN, Decision Tree, Logistic Regression, Naïve Bayesian
Dalianis (2018)	Forecasting technology trends	Clustering
Salloum, Al-Emran, Monem & Shaalan (2018)	Extracting Information from Research Articles	Classification, clustering, summarization, and topic detection
Sun, Cai, Li, Liu, Fang, & Wang (2018)	Extracting information on electronic medical record	CNN and CRF model
Wu & Lin (2018)	Retrieving E-Commerce Logistics knowledge	Association Rules
Moloshnikov, Sboev, Rybka, & Gydovskikh (2015)	Finding documents on a particular topic depending on a selected reference collection of documents	Word cloud, association, clustering, word frequency

2.4 Summary

Upon reading each of the authors' work depending on the requirement needed on the extraction various text mining technique can be applied. The most similar model that is required after reading the research is BERT and CRF thus I will consider both techniques in the modeling phase however if the objective that was stated wasn’t met or the goal wasn’t reached, other possible data mining techniques will be considered for a comparison purpose.

CHAPTER 3: TECHNICAL RESEARCH

3.1 Programming language chosen

R and Python are open-source programming languages with a massive community. New libraries are added continuously to their respective catalog. R is used for statistical assessment at the identical time as Python gives a more favored approach to statistics science. R and Python are great in terms of programming language oriented in the direction of statistics science. Learning both languages is the perfect solution, but it is time-consuming. Python is a favored-motive language with a readable syntax. R, however, is built thru manner of the method of statisticians and encompasses their language (Johnson, 2022).

3.1.1 R Language

Scholars and statisticians had been growing R for over 20 years. R is now one of the richest languages for information analytics. The Open-Source Repository (CRAN) has approximately 12,000 packages. You can locate libraries for any evaluation you need to perform. The library makes R the first-class desire for statistical evaluation, specifically for specialized analytical tasks. The maximum superior distinction among the R and different statistical merchandise is the output. R has superb reporting tools. RStudio comes with the Knitr library (Fiducia, 2022).

3.1.2 Python

Python can do lots of the equal components as R: information processing, development, feature-selective net scraping, applications, and extra. Python is a device for deploying and enforcing device mastering at scale. Python code is less difficult to keep and extra dependable than R. a few years ago; Python didn`t have many libraries for an information evaluation and device mastering. Recently, Python is catching up and giving cutting-edge APIs for device mastering or Artificial Intelligence. Most of the information technology activity may be executed with 5 Python libraries: Numpy, Pandas, Scipy, Scikitlearn, and Seaborn. Python, on the alternative hand, makes

replicability and accessibility less difficult than R., in case you want to apply the effects of your evaluation in software or website, Python is the first-class desire (IBM, 2021).

3.1.3 Summary

For the project, since I'm planning to do a data-science-related project, as well as building a machine learning algorithm relying on libraries, I'll be using phyton as it is a more suited language to be used.

3.2. IDE Chosen

3.2.1. Machine Learning Development (Jupyter Notebook)

Google Colab is a free IDE available on the internet. It is a product of Google studies and is primarily based totally on Jupyter. Colab is a brilliant device for novice and expert users, nearly all essential libraries are preinstalled with it so that there's no need to deploy libraries one by one. Colab`s notebook documents are saved for your google drive, so that they may be accessed from everywhere you want. It additionally permits you to percentage your pocketbook together along with your colleague without even downloading it, which is significantly the high-quality function for many. Apart from this it additionally offers loose GPU and TPU in your paintings and that makes it perfect for Deep Learning and Machine Learning projects. The main advantage of Google Colab is its convenience of it and does not require tons of resources to run the code as all the resources are provided by google cloud service. Other than that, Colab comes with the libraries thus it will save time from installing libraries (Orhan, 2020).

In comparison with Jupyter Notebook, it allows you to use RAM, CPU, and GPU that's available of your hardware. It only allows limited RAM and CPU, meaning if the code require long processing will take a while for it to complete on Google Colab. Which is why the author decided to switch from Google Colab to Jupyter Notebook as it allows the author to take advantage of its own hardware that has high specs.

3.2.2. Website Development (PyCharm)

The purpose of PyCharm is to offer a comprehensive solution for developing whole Python packages and software, including classes and graphical user interfaces (GUIs). It also performs well in complex contexts where it is necessary to manage the interactions between numerous scripts. A built-in debugger, intelligent auto-complete, and DevOps capabilities like version control are among of PyCharm's most well-liked features, which make it the perfect tool for programmers and software engineers. For website deployment PyCharm will be used to deploy the application as the application file runs on ‘py’ instead of ‘ipynb’.

3.3. Libraries chosen / Tools chosen

3.3.1. pandas

Pandas is the most popular open-source Python package for data processing/data analysis and machine learning tasks. It is built on top of another package called Numpy which supports multidimensional arrays. As one of the most popular data processing packages, Pandas works well with many other data processing modules in the Python ecosystem and is included in every Python distribution, from those that usually ship with operating systems to commercial vendor distributions like ActiveState's ActivePython (Rashi, 2019).

3.3.2. string

Python String module contains some constants, utility function, and classes for string manipulation. Since the project involved textual data, the string library will be used to manipulate the textual data, such as removing whitespace, punctuation, or separating the string (JournalDev, 2021).

3.3.3. tqdm

Tqdm is a library in Python that's used for developing Progress Meters or Progress Bars. tqdm were given its call from the Arabic call taqaddum which means `progress`. Implementing tqdm may be completed results easily in our loops, capabilities or maybe Pandas. Progress bars are quite beneficial in Python as it permits to look if the Kernel continues to be running and Progress Bars are visually attractive to the eyes. Other than that, it offers Code Execution Time and Estimated Time for the code to finish which might assist at the same time as running on large datasets (Shah, 2021)

3.3.4. random

The random module of the phyton library allows them to generate random numbers. The random number is generated according to the generator's seed. This module can be used to perform random operations such as generating random numbers, printing random values for lists or strings, etc (Phyton, 2022).

3.3.5. Json

The JSON module is specifically used to transform the python dictionary above right into a JSON string that may be written right into a file. While the JSON module will convert strings to Python data types, typically the JSON capabilities are used to examine and write without delay from JSON files (Navone, 2020).

3.3.6. os

The OS module in Python affords features for growing and casting off a directory (folder), fetching its contents, converting, and figuring out the modern-day directory, etc. This module offers a transportable manner of the use of running device structured functionality (TutorialsTeacher, 2022).

3.3.7. ast

The ast module helps Python applications handle Python abstract syntax grammar trees. The abstract syntax itself may change from release to release of Python. This module helps you to know programmatically what the current grammar looks like.

3.3.8. spaCy

SpaCy is a Python NLP library that is open-source and free. It is created to produce information extraction or natural language processing systems and is built in. It offers a clear and approachable API and is designed for use in production. It includes word vectors, NER, POS tagging, dependency parsing, and other features (Singh, 2021).

3.3.9. scikit-learn

The most effective and reliable Python machine learning library is called Sk-Learn. Through Python consistency interface, it offers a variety of effective tools for statistical modelling and machine learning, including classification, regression, clustering, and dimensionality reduction.

3.3.10. Flask

Flask module is a web framework that offers the technology, tools, and libraries necessary to create a web application. This web application may consist of a few web pages, a blog, a wiki, or it may be as large as an online calendar or a for-profit website (Python, 2022).

3.3.11. matplotlib

For Python and its numerical extension NumPy, Matplotlib is a cross-platform data visualisation and graphical charting package. As a result, it presents a strong open source substitute for MATLAB. The APIs (Application Programming Interfaces) for matplotlib allow programmers to incorporate graphs into GUI applications (ActiveState, 2022).

3.3.12. SciPy

For many different types of problems, including those involving algebraic equations, differential equations, statistics, eigenvalue issues, integration, interpolation, and optimization, SciPy offers algorithms. The SciPy data structures and algorithms have broad domain compatibility (SciPy, 2022).

3.3.13 Seaborn

Python's Seaborn package allows you to create statistical visuals. It incorporates tightly with Panda’s data structures and is built upon Matplotlib. You may examine and comprehend your data with Seaborn. Its charting functions work with data frames and arrays that include entire datasets, and they internally carry out the semantic mapping and statistical aggregation required to make useful graphs. You can concentrate on what the various components of your plots represent rather than the specifics of how to draw them thanks to its dataset-oriented, declarative API (seaborn, 2022).

3.4 Operating System chosen

Windows 10 is the operating system chosen for the project as it supports most of the applications available. Especially since I'm using Windows 10 as the default operating system, and Google Colab as IDE, Windows 10 is a great choice as Google Colab is web-based and does not have strict requirements (scikit, 2022).

3.4.1 Hardware Requirement

Following are minimum requirement to run Jupyter Notebook:

Table 3.1: Hardware Requirement

Name	Requirement	Current
Processor	Intel Core i5-4590	AMD Ryzen 9 4900H
Graphics Card	NVIDIA GeForce GTX 970	NVIDIA GeForce RTX 2060
Memory	4 GB	16 GB
Available Space	2 GB	100+ GB

3.4.2 Software Requirement

Following are minimum requirement to run Jupyter Notebook:

Table 3.2: Software Requirement

Name	Requirement	Current
Operating System	Windows 7	Windows 10
Software	Any Internet Browser	Google Chrome

3.5 Web Server chosen

Flask is a Python web framework made for deploying simple and clean, pragmatic design. Built through skilled developers, it looks after lots of the problems of net improvement, so that you can recognize on writing your app while not having to reinvent the wheel. It`s loose and open source.

3.6 Web browser chosen

Chrome is designed to be the quickest internet browser. With one click, it masses internet pages, a couple of tabs, and packages with lightning speed. Chrome is equipped with V8, a quicker and extra effective JavaScript engine. Chrome additionally masses internet pages quicker using the WebKit open supply rendering engine.

3.7 Summary

Phyton is the chosen programming language and the IDE chosen is Jupyter Notebook. The libraries chosen included Pandas, string, tqdm, random, os, ast, spaCy, scikit-learn, Flask, SciPy, and Seaborn. The chosen operating system is Windows 10 as Google collab has no strict requirement and it is my default operating system. The web server chosen is Django a high-level net framework, and the web browser chosen is Google Chrome as it is very efficient to use it.

CHAPTER 4: METHODOLOGY

4.1 Introduction

The methodology that can be chosen is between SEMMA, CRISP-DM, and KDD. The SEMMA stands for Sample, Explore, Modify, Model, and Assess. It is a listing of sequential steps evolved through SAS Institute, certainly considered one among the most important manufacturers of facts and enterprise intelligence software. CRISP-DM stands for Cross Industry Standard Process for Data Mining, it is almost the same as SEMMA, but it has six stages that describe the records technological know-how existence cycle where it includes business understanding and deployment of the system. The KDD stands for Knowledge discovery in database, where it consist seven stages.

4.1.1 Comparison Analysis of SEMMA, CRISP-DM & KDD

4.1.1.1 CRISP-DM

CRISP-DM defines a framework for specifying a data mining project and establishes the activities that must be performed to complete a product or service. The task consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. This method is inexpensive because it requires multiple processes to solve a simple data mining problem and moreover. CRISP-DM promotes excellence and allows project duplication and most importantly, this methodology provides an integrated framework for project planning and management without needing to be concern about the discipline as it can be implemented in most data science projects (Wirth & Hipp, 2000).

4.1.1.2 SEMMA

SEMMA stands for Sample, Explore, Modify, Assess. SAS Institute, which developed the model, describes it as not a data mining tool method but a set of tools to perform essential data tasks mining. SEMMA focuses most on the model development aspects of data mining and is used in

the SAS Enterprise Mine software. the movement between different phases are not tight, during the project you can move back and forth and repeat the steps (SAS Institute, 2017).

4.1.1.3 KDD

Knowledge discovery in databases (KDD) is the process of discovering interesting and useful knowledge in databases. Although it may seem like data mining, data mining is just one step in the KDD process where algorithms are applied to find patterns in data. KDD focuses on the entire process of extracting knowledge from data, including how data is stored and accessed, how algorithms perform efficiently while being used on large data sets, and how to interpret and visualize the results. The other steps in this process are designed to extract useful knowledge from data (Fayyad, Piatesky-Shapiro, & Smyth, 1996).

4.1.2 Summary

Table: 4.1 Summarized Strengths and Disadvantages
(Daderman & Rosander, 2018)

Framework	Documentation	Strengths	Weakness
CRISP-DM	There is website available where it is easy to be understood as the research is based on research papers or journal	• Clearly Define Process and fully prepared for upcoming problems • Suitable for large project • Flexible with most of the data mining techniques • Iterative method	• The process are complicated • Data preparation and modeling are different from traditional static data.

		• Documentation can easily be found	
KDD	There is no website available. The description of this framework solely based on research paper and requires background in data mining field	• Iterative Method • Support most of the data mining technique	• Require experience and knowledge of data mining
SEMMA	There is website available that was created by SAS Institute. The full domcumentation are all available for free and solely focus on the framework knowledge	• Detailed information provided from the documentation • Support most of the data mining technique • Iterative method	• It is designed tow work with SAS Enterprise Miner toll • Business Understanding phase was not included

Following table 4.1, an analysis of the table is summarized. Based on the information from the table, the most suitable methodology would be CRISP-DM as the topic that was focused on was related to business, specifically E-Commerce, thus the business understanding phase was required and needed to be included. Other than that, CRISP-DM is suitable for me as the knowledge of data mining that I possess is not huge enough to call myself an expert, thus it is concluded CRISP-DM is the best methodology for this project

4.2 Selected Methodology (CRISP-DM)

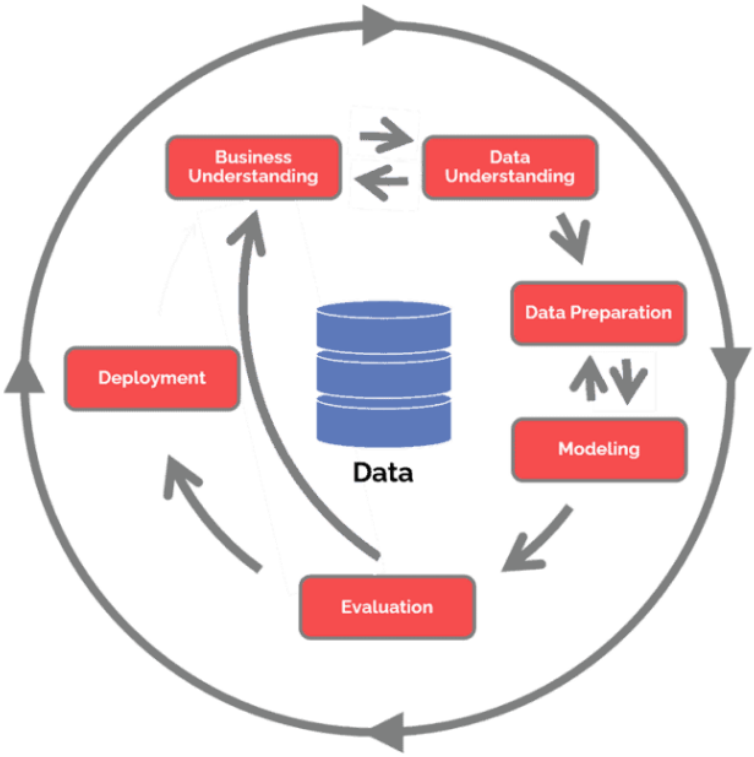


Figure 4.1: CRISP-DM Diagram

(Hotz, 2022)

The base methodology that is chosen is CRISP-DM. As shown in Figure 4.1, the diagram shows that there are six stages in the methodology where it allows the looping or rollback whenever the stage reached the evaluation phase, and the result is not the desired or has yet reach the requirement.

4.2.1 Business Understanding

In the business understanding stage, first, we must understand and determine the business objectives which can help assess the situation do determine the data mining goals and produce a detailed project plan. The business objectives for the project are to be able to determine address is incorrect or correct, and if it is correct, the address would be able to be categorized between the street address and the destination. These are all possible even when the customer provided the

address in unstructured form. The data mining goal is to define the business objective which is to develop a machine learning algorithm that has a decent accuracy rate to predict, extract, and label the incorrect address. With more context, the element extraction allows the restriction of passing the correct address only where the model will label the correct one into a token of '1' and the wrong one will be labeled with a token of '0' thus will be returned to the customer as incorrect. This is to increase effectiveness when processing address of whether it is correct or incorrect. Apart from just determining and extracting information out of the raw address, the collected address can also be passed into geocoding which can be processed into many things. The data that was gathered was already standardized thus allowing the analysis team to easily process the data and get useful insights regarding the geographics of the customer for much easier visualization and reading.

4.2.2 Data Understanding

To train the model according to the business objectives that were provided, the data that we rely on was collected from Kaggle, an open-source website. However, the data that was gathered was trusted, as it is directly sourced from Shopee of Indonesia, an E-commerce company that was highly popular and used in Indonesia. According to SimilarWeb (2022), the number of visits to the website reached over 100 million.

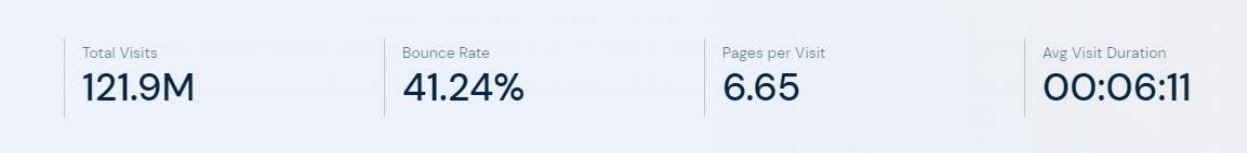


Figure 4.2: KPI of Shopee in Indonesia
(SimilarWeb, 2022)

4.2.2.1 Basic Text Data Pre-processing and Data Cleaning

The data that was gathered was directly separated into test data and train data where the test data contain 50000 rows of data with 2 variables (id, raw_address), and the train data 300000 rows of

data with 3 variables (id, raw_address, POI/street). The sample of the data can be seen from the figure below.

id	raw_addre	POI/street
0	jl kapuk tin	/jl kapuk timur delta sili iii lippo cika
1	aye, jati sa	/
2	setu siung	/siung
3	toko dita,	toko dita/
4	jl. orde bai	/jl. orde baru
5	raya samb	toko bb kids/raya samb gede
6	kem mel ra	/kem mel raya
7	tela keurai	/tela
8	gg. i wates	/gg. i
9	bunga nco	/bunga ncole ix
10	cikahuripa	sd negeri bojong 02/klap boj
11	yaya atoha	yayasan atohariyah/
12	abim ix 24	/abim ix
13	gang xiii ru	/gang xiii
14	kamp utan	/
15	kampung.g	gudang areng/
16	maru baru	/maru baru 2

Figure 4.3: Sample of the Train dataset

Normally, we would have to filter the information since the only information we needed is the raw address, since Kaggle has already preprocessed the data and gotten rid of the information that is not needed such as customer name, customer phone number, the product bought, time bought, etc., in other words, information was already filtered thus there is no need to filter the data and leave it as it is. The next phase is to check whether there is missing data, any null value, or duplicate value that is available in the dataset.

	Total	Percentage
POI	178509	59.50
street	70143	23.38
POI/street	31993	10.66
id	0	0.00
raw_address	0	0.00

	Columns	Duplicate count
0	id	0
1	raw_address	0
2	POI/street	4065
3	POI	12422
4	street	50254

Figure 4.4: Missing and Duplicate Values Percentage

The most important part of of this phase is checking the raw_address as for that variable there cannot be any missing or duplicate value. According to figure 4.4, there are no missing values and duplicate values for raw_address which is good, thus there is no need to drop any data. For other variables, the missing values and duplicate values show that there is a pattern and there are mistakes made by the customers which are useful when training the machine learning algorithm in the later phase. Thus, these missing and duplicate values will not be altered and will be left alone.

4.2.2.2 Data Exploration for Text Data

The data will be explored using N-gram Model, to determine the most used words and their pairings. Text length distribution can also be used to explore the data to discover the length distribution of the address. Semantic Analysis can also be performed to find out the percentage of question marks, full stops, commas, capital letters, and numbers included in the address.

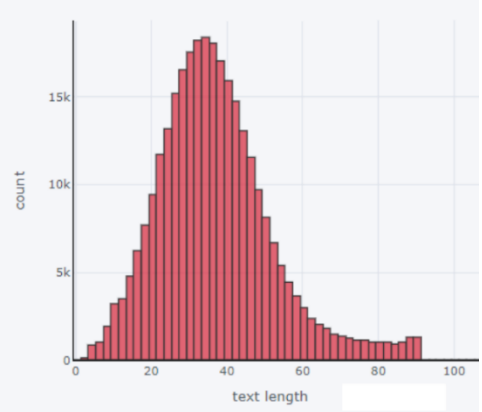


Figure 4.5: Text Length Distribution

```
Address with question marks: 0.01%
Address with full stops: 19.77%
Address with comma: 50.68%
Address with capital letters: 0.00%
Address with numbers: 59.32%
```

Figure 4.6: Semantic Analysis

4.2.3 Data Preparation

In the preparation phase, the data should be selected thoroughly, cleaned to remove NULL values, and should be reconstructed if needed, as well as integrated to create new data sets from other sources. Lastly, if integration was done reformatting the data should be done too as the data should be standardized instead of having multiple formatting.

Through the verification of the data from the data understanding phase, it was verified that the data was clean and does not require to be integrated, or to be cleaned. The main reason there's no need to be cleaned is that the data is mainly unstructured textual data thus cleaning was not needed. Though the data will be reformatted into tokens for the tokenization phase which will be explained in detail. The other possibility is building word lists which are to categorize the address. The following table is the example of extraction of POI and Street from the data which are possible to be used when building the model.

Table 4.2: Sample data

id	raw_address	POI/street	POI	street
0	raya samb gede, 299 toko bb kids	toko bb kids/raya samb gede	toko bb kids	raya samb gede
1	aye, jati sampurna	/	NA	NA
2	setu siung 119 rt 5 1 13880 cipayung	/siung	NA	siung

4.2.3.1 Tokenization and Labelling for Named Entity Recognition Task

Named entity recognition (NER), additionally known as entity chunking, identity, or extraction, is the challenge of detecting and classifying key information (entities) in the textual content. In different words (Mansouri, Affendey, & Mamat, 2008), a NER version takes a bit of textual content as entering and for every phrase withinside the textual content, the version identifies a

class the phrase belongs to. The raw_address will be divided into multiple tokens per sentence, The first token will be used as a reference token for the prediction and since all the tokens are connected via self-attention there will be no problem for predicting the tokens of the words. The labeling will be categorized in a detailed manner to increase the accuracy when building the model. The tagging format that will be used will be IOB tags which is a common tagging method to tag chunks of tokens.

4.2.4 Modeling

4.2.4.1 Selecting Modeling Techniques

In this phase, multiple algorithms should be prepared and discussed. The algorithm chosen will be based on research, but the prepared list of algorithms is there just in case the business objective was not met, and the goal is not fulfilled thus the phase will be restarted back to business understanding and the model selection will be conducted again to develop a better model. For better context, the model that will be discussed is BERT and CRF model. The BERT model is chosen to develop NER and the CRF model is to develop a spelling correction model.

4.2.4.2 Generate Test Design

In this stage, data are usually segmented for better accuracy when developing the model. In this phase, data are processed into three sets of data which usually are 80% training data, with 10% validation data, and 10% set (Fazel, 2021; Agrawal, 2021). The data that was gathered from the Kaggle website, was already split. Thus, for the test design, since the data are already split into different sets, which is train data and test data there’s no need to generate another test design unless the generated test design is not suitable thus the data need to be revised accordingly.

4.2.4.3 Assessing the Model

The selection of the model will be depending on the accuracy score of the model itself where it is measured by the number of correct predictions. The following is the accuracy metric.

$$accuracy((p_i, s_i), (\hat{p}_i, \hat{s}_i)) = \begin{cases} 1 \ if \ p_i == \hat{p}_i \ and \ s_i == \hat{s}_i \\ 0 \ otherwise \end{cases}$$

Where:

p_i = the actual POI name for ith address
$\hat{p}_i$ = the predicted POI name for ith address
s_i = the actual street name for ith address
$\hat{s}_i$ = the predicted street name for ith address

The addresses are often followed by missing POI or street elements thus for such situations, the specific element should be left empty. The following formula is to calculate the average accuracy score of the model

$$score = \frac{1}{n} \sum_{i=1}^{n} (accuracy((p_i, \hat{p}_i), (s_i, \hat{s}_i)))$$

Where:

n = the total number of addresses
$accuracy$ = the function provided above

4.2.4.4 Building the Model

The building approach for the prediction, the first stage would be a token classification task, which is the necessary starting stage when executing the BERT model.

Token classification Example

- Raw_address: raya samb gede, 299 toko bb kids
- POI : toko bb kids
- Street: raya samb gede
- Output: ["raya", "samb", "gede", ",", "299","toko","bb","kids"]

After the token classification task is done, the next step is to convert the data into BERT-readable token IDs. After the implementation of the tokenizer, the dataset will be framed as an extractive Question and Answer problem where context, which is the tokenized data, are provided to predict the POI and Street by giving the range of token that is the element depending on the raw_address that is provided.

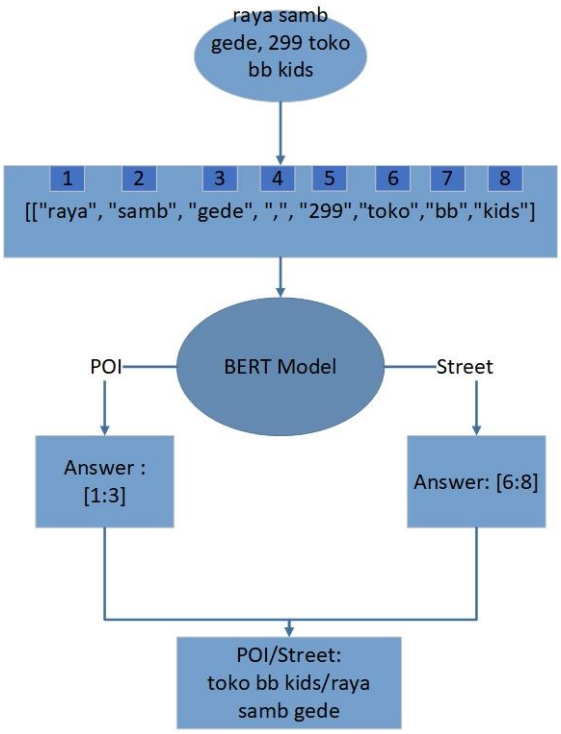


Figure 4.7: Overview of Name Entity Recognition Model

4.2.4.5 Spelling Correction Model

Based on the dataset's raw addresses, POI, and Street the first step is to separate the incomplete and complete addresses by labeling them for example for the complete address will be labeled as 1 and the uncompleted one will be labeled as 0.

Table 4.3: Sample data when included labelling

id	raw_address	POI/street	Label
0	yaya atohar,	yayasan atohariyah/	0
1	jl. amd, komplek borneo lestari, blok 2, no. 30	komplek borneo lestari/jl. amd	1
2	toko bang ajs,	toko bangunan ajs/	0

After the labeling is done, based on the label along with complete POI/Street pairs in the training set, the model can be built by mapping between incomplete components, and it's restored abbreviation. For the table 4 the incomplete component would be:

- yaya = yayasan
- atohar = atohariyah
- bang = bangunan

After building the dictionary, it can be used for restoring incomplete addresses, and to do this, the most efficient way would be adding the 2-gram, and 3-gram models to be applied in the model to know the pattern of subsequent words. With the help of a 2-gram and 3-gram model and instead of just relying on BERT model, it will be much more accurate as Indonesian language is complex.

Example:

- Incomplete word: "indone”
- Full Forms Possibility: ["indonesia.", "indonesian", "indonesiaku", …]

With the help of 2-gram and 3-gram, the pattern of the most frequent paired word will be known thus prediction will be more accurate than just relying on the BERT model itself

4.2.5 Evaluation

In this phase, the trained model will be tested with the real generated data where the success criteria will be depending on the business objectives that were determined in the business understanding phase. The model built will be assessed based on the accuracy metric above. Since this project only includes one set of models the evaluation will be based on only accuracy instead of comparing between models. According to Vallantin (2018), 70% - 80% is the base accuracy the model should reach to meet business requirements. Following the requirement, the model will be considered a success when the accuracy has reached more than 70%. If it doesn’t meet the requirement, it is possible to return to the business understanding phase and re-analyze the problems.

4.2.6 Deployment

The final stage is where it involves deploying the model into the real-world environment. Thus, proper deployment planning needs to be thought of first, the strategy to deploy the model and the steps to deploy are stated in the planning concisely. After the planning is finished, there will also be a monitoring and maintenance stage, summarizing the strategy of the actions along with the steps, this is to ensure the model will work as the way the model was intended to during the modeling phase. In this phase, I will be using Django as the web server for the deployment, where the system is able to allow input and able to show output. The input will be entered through the website and be processed by the model to produce the extraction of point of interest and street address, as well as the label.

4.3 Summary

The methodology that is used is CRISP-DM, where the process started with business understanding to understand the main objective of the project which is the extraction of the address from the raw address obtained from the customer. After that in Data understanding, the data is collected from Shopee Indonesia through the Kaggle website, and it is described and explored. The quality does not need to be confirmed as it is meant to be unstructured. Next is the data preparation phase where data are formatted into a token for it to be readable by the BERT model. The data is not clean, selected, and constructed mainly because it is unstructured data. In the modeling phase, BERT, CRF, and N-gram models are chosen. BERT models are used to build the NRE model, and the CRF and N-gram models are used to build the spelling correction model. For the evaluation, the accuracy metric is provided to determine the plan for deployment. The deployment phase will involve Django as the webserver where the website is available to enter input and provide the output of the prediction and extraction.

CHAPTER 5: DATA ANALYSIS

5.1. Introduction

In the Data Analysis section data collection methods will be explained and evaluated whether the method of data collection is justified as well as understanding and exploring the data by using Phyton to get the gist how the data is structured and how the data will be processed for it to be passed for model building. After the data is processed some screenshot will be provided to further explain the dataset and the model that will be used for the data set will be based on the findings of data understanding.

5.2. Data Collection

Data Collection methodology depending on the aim of research and type of data that you want to collect to achieve the goal of the research. There are various methods of collecting the data but to collect high quality data which are accurate and as similar as real life situation, the dataset will be collected from a Indonesia E-Commerce Company that require the customer to manually enter address for product collection/destination. Thus, concluding from that the method suitable for data collection would be online observation. There are two requirement of the dataset that is to be collected, the first one would be containing the raw address entered by customer or user and next requirement would be having the correct address that should be entered by the customer for evaluation purposes. The conclusion method that was deployed to collect dataset required for this research is from Kaggle, website used by data scientists and data analysts for dataset publishing and data collection. The dataset that was chosen is collected from Shopee which includes test.csv (50000 rows) and train.csv (300000 rows).

```
df.head()
```

	id	raw_address	POI/street
0	0	jl kapuk timur delta sili iii lippo cika 11 a ...	/jl kapuk timur delta sili iii lippo cika
1	1	aye, jati sampurna	/
2	2	setu siung 119 rt 5 1 13880 cipayung	/siung
3	3	toko dita, kertosono	toko dita/
4	4	jl. orde baru	/jl. orde baru

```
[ ] validation.head()
```

	id	raw_address
0	0	s. par 53 sidanegara 4 cilacap tengah
1	1	angg per, baloi indah kel. lubuk baja
2	2	asma laun, mand imog,
3	3	ud agung rej, raya nga sri wedari karanganyar
4	4	cut mutia, 35 baiturrahman

Figure 5.1: Sample Test and Train dataset

In order to get useful insights from the visualization below, the components explained below will help understanding an Indonesian address better.

From the highest to the lowest:

- Provinsi — Province
- Kota or Kabupaten — Regency
- Kecamatan — District
- Kelurahan or Desa — Village
- RW, an abbrev. of Rukun Warga (Neighborhood Unit)
- RT, an abbrev. of Rukun Tetangga (Community Unit)

The last two are subdivisions of a Village, and normally use numbers.

Table 5.1. Address Simplification Example

Example 1
Jl.janaka Rt 1 Rw 1 krajan Wringinanom Sambit Kabupaten Ponorogo Jawa Timur • Jl Janaka — street name • RT 1, RW 1 — as explained above • Krajan — Village • Wringinanom — District • Kabupaten Ponorogo — Regency • Jawa Timur — Province
Example 2
Jalan Candi Panggung Barat. No 16 . RT 01 RW 18. Kelurahan Mojolangu, Kecamatan Lowokwaru Malang City , East Java • Jalan Candi Panggung Barat. No 16 — street name and number • RT 01, RW 18 — as explained above • Kelurahan Mojolangu — Village • Kecamatan Lowokwaru — District • Malang City — Regency • East Java (Jawa Timur) — Province

Commonly used abbreviation for Indonesian Address:

- gg/gg./gang referes to an alley.
- jl/jln/jalan/jl./jln. refers to Jalan (Street)
- no./nomor/no refers to number
- kec./kecamatan/kec refers to Kecamatan
- kel./kelurahan/kel refers to Kelurahan
- kab./kabupaten/kab refers to Kabupaten

5.3. Data Understanding/Exploration

Main purpose of data exploration is to find out unknown insights efficiently usually when this phase was conducted the knowledge that was found has no purpose yet as the analyst are unsure of what they are looking for (Idreos, Papaemmanouil & Chaudnuri, 2015). In this exploration phase basic data preparation and exploration will be done along with some analysis to view the data in-depth. The analysis method would be semantic analysis, text length analysis. The distribution of the data will also be analyzed by using top unigram distribution, top bigram distribution, and top Trigram distribution, where all the distribution methods mentioned will be provided with visualization to get proper view of the dataset distribution. Additional charts will also be provided including Most common words as well as word cloud.

5.3.1. Data Preparation

```
!pip install chart_studio
!pip install textstat

import numpy as np
import pandas as pd

# Visualization
import matplotlib.pyplot as plt
import chart_studio.plotly as py
import plotly.figure_factory as ff
import plotly.graph_objects as go
from plotly.offline import iplot
import cufflinks
cufflinks.go_offline()
cufflinks.set_config_file(world_readable=True, theme='pearl')
import seaborn as sns
%matplotlib inline

# Word Cloud
from wordcloud import WordCloud

# sklearn
from sklearn.feature_extraction.text import CountVectorizer

# Pandas Profiling
from pandas_profiling import ProfileReport

# Suppress warnings
import warnings
warnings.filterwarnings('ignore')
```

Figure 5.2: Library Imports

The preparation starts of with importing all of the necessary library that is needed for data understanding and data exploration. The libraries that was used for data exploration would be numpy, pandas, pyplot, plotly, cufflinks, seaborn, wordcloud, sklearn, and warnings. Some of the library are not necessarily needed to imported for example the warnings library are not necessary but for better visualization of the screenshots the warning will be ignored thus the warnings library is used in this case for that scenario.

5.3.2. Basic Data Exploration

The data exploration is conducted with Google Colab as it is much more efficient, and it is a resource saving method. The data exploration started off setting up data frame so that it is easier to explore the dataset.

```
df[['POI', 'street']] = df['POI/street'].str.split('/', 1, expand=True)
df['POI/street'].replace('/', np.nan, inplace=True)
df['POI'].replace('', np.nan, inplace=True)
df['street'].replace('', np.nan, inplace=True)
df.head()
```

	id	raw_address	POI/street	POI	street
0	0	jl kapuk timur delta sili iii lippo cika 11 a ...	/jl kapuk timur delta sili iii lippo cika	NaN	jl kapuk timur delta sili iii lippo cika
1	1	aye, jati sampurna	NaN	NaN	NaN
2	2	setu siung 119 rt 5 1 13880 cipayung	/siung	NaN	siung
3	3	toko dita, kertosono	toko dita/	toko dita	NaN
4	4	jl. orde baru	/jl. orde baru	NaN	jl. orde baru

Figure 5.3: Splitting POI and Street individually

The code shown on Figure 5.3 allows you to split the row of “POI/street” into two parts individually which is POI and street. This allows the developer to look at the POI and street as an individual variable rather than as group. The code also included to replace the missing values to NaN which allow us to analyze it in the next step which is to check the missing values out. The dataset that is used for data exploration are only the test set as it has a greater number of rows and it has the data we need to analyze which is the POI/street.

```
def missing_value_of_data(data):
    total = data.isnull().sum().sort_values(ascending=False)
    percentage = round(total/data.shape[0]*100,2)
    return pd.concat([total,percentage], axis=1, keys=['Total','Percentage'])
```

```
missing_value_of_data(df)
```

	Total	Percentage
POI	80332	59.57
street	31496	23.35
POI/street	14402	10.68
id	0	0.00
raw_address	0	0.00

Figure 5.4: Missing values

The missing values as shown in figure 5.4. There are around 10.7% of missing values for POI/Street but as individual variable, the POI has 59.6% missing value, and the street has 23.3% missing values. One of the reasons why the POI/Street missing is because the customer entered the raw_address not as intended thus nothing useful can be extracted out of the raw_address. The same applies to POI and Street the missing values are the result from raw_address not having Point of interest or street name stated in the raw_address. This shows the percentage of customer entering wrong address. The missing values will not be removed as it allows the model building to learn what to extract out of the raw_address and what to not extract. The main reason is because the are high amount of missing values thus if it was removed it will affect the dataset.

```
def duplicated_values_data(data):
    dup=[]
    columns=data.columns
    for i in data.columns:
        dup.append(sum(data[i].duplicated()))
    return pd.concat([pd.Series(columns),pd.Series(dup)],axis=1,keys=['Columns','Duplicate count'])
```

```
df_cleaned = df.dropna()
duplicated_values_data(df_cleaned)
```

	Columns	Duplicate count
0	id	0
1	raw_address	0
2	POI/street	573
3	POI	2227
4	street	9767

Figure 5.5: Duplicate values

The duplicate values are shown in Figure 5.5. There are only 573 duplicate values out of 300000 of the rows. The POI have 2227 duplicates and street has 9767 duplicates. This means that the customer of the duplicate values are the same or they simply living in the same street either way for model building to achieve higher accuracy the duplicate values will be removed as the number of the duplicate are low thus will not affect the dataset that much.

5.3.3. Semantic Analysis

```
qmarks = np.mean(df['raw_address'].apply(lambda x: '?' in x))
fullstop = np.mean(df['raw_address'].apply(lambda x: '.' in x))
comma = np.mean(df['raw_address'].apply(lambda x: ',' in x))
capital_first = np.mean(df['raw_address'].apply(lambda x: x[0].isupper()))
capitals = np.mean(df['raw_address'].apply(lambda x: max([y.isupper() for y in x])))
numbers = np.mean(df['raw_address'].apply(lambda x: max([y.isdigit() for y in x])))

print('Address with question marks: {:.2f}%'.format(qmarks * 100))
print('Address with full stops: {:.2f}%'.format(fullstop * 100))
print('Address with comma: {:.2f}%'.format(comma * 100))
print('Address with capital letters: {:.2f}%'.format(capitals * 100))
print('Address with numbers: {:.2f}%'.format(numbers * 100))
```

```
Address with question marks: 0.02%
Address with full stops: 19.79%
Address with comma: 50.79%
Address with capital letters: 0.00%
Address with numbers: 59.12%
```

Figure 5.6: Semantic Analysis

Semantic analysis allows us to know the general structure of the raw_address build of. For example, as shown in Figure 5.6 it allows us to know that there are some question mark included in the raw_address which will be removed as if it was kept it will be hindrance for model building. The numbers should also be removed as RT and RW as well as road number which contributes to most of the number are not needed to be included in POI/Street. This also applies to comma and full stops symbols that customer typed in when giving their address, the symbols will later on be filtered for model building.

5.3.4. Text Length Analysis

Text Length analysis allow the developer to get insight on the unstructured text for model building, knowing whether the text is short or long overall. According to Amplayo and the team (2019), text length are necessary to be analyzed as tested text classification modelling depending on the text length, the dataset could be generalized depending on the length of text thus much easier to predict. Amplayo and his team separated reviews into two types which is the short review and the long review which allow them to gather insight much easier.

```
lens = df.raw_address.str.split().apply(lambda x: len(x))
print(lens.describe())
```

```
count    84825.000000
mean         6.836475
std          2.834046
min          1.000000
25%          5.000000
50%          6.000000
75%          9.000000
max         32.000000
Name: raw_address, dtype: float64
```

Figure 5.7: Text Length Analysis

```
df['text_len'] = df['raw_address'].astype(str).apply(len)
df['text_word_count'] = df['raw_address'].apply(lambda x: len(str(x).split()))
df.head()
```

	id	raw_address	POI/street	POI	street	text_len	text_word_count
0	0	jl kapuk timur delta sili iii lippo cika 11 a ...	/jl kapuk timur delta sili iii lippo cika	NaN	jl kapuk timur delta sili iii lippo cika	66	13
1	1	aye, jati sampurna	NaN	NaN	NaN	18	3
2	2	setu siung 119 rt 5 1 13880 cipayung	/siung	NaN	siung	36	8
3	3	toko dita, kertosono	toko dita/	toko dita	NaN	20	3
4	4	jl. orde baru	/jl. orde baru	NaN	jl. orde baru	13	3

Figure 5.8: Text Length Data Frame

Figure 5.7 shows that raw_address on average can be concluded as short text as it has the mean of 7 count of words. Though there are 32 count of words it still not considered as long text. For better visualization of text length distribution dataframe is shown on Figure 5.8, it was included two extra rows of variable which is text_len and text_word_count. The “text_len” derived from the length

of the text where one character is considered as one where as the “text_word_count” is derived from the number of words in the text.

```
[38] enable_plotly_in_cell()
     df['text_len'].iplot(
         kind='hist',
         bins=100,
         xTitle='text length',
         linecolor='black',
         color='red',
         yTitle='count',
         title='Text Length Distribution')
```

Figure 5.9: Text Length Distribution iplot Code

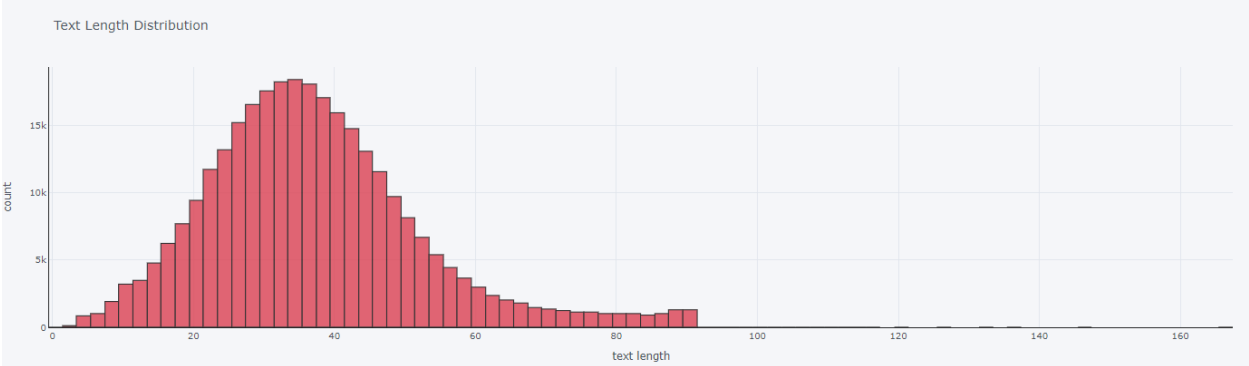


Figure 5.10: Text Length Distribution Visualization

There’s also text length distribution visualization which is shown on Figure 5.9 and 5.10 which allow us to see the distribution of the dataset based on “text_len”. On Figure 5.12 it visualizes the distribution of “text_word_count”. This shows that the average text length is around 35 and some of the text length reached more than 80 words. This means Indonesian address

```
[18] enable_plotly_in_cell()
     df['text_word_count'].iplot(
         kind='hist',
         bins=50,
         xTitle='text word count',
         linecolor='black',
         color='red',
         yTitle='count',
         title='Text Word Count Distribution')
```

Figure 5.11: Text Word Count Distribution iplot Code

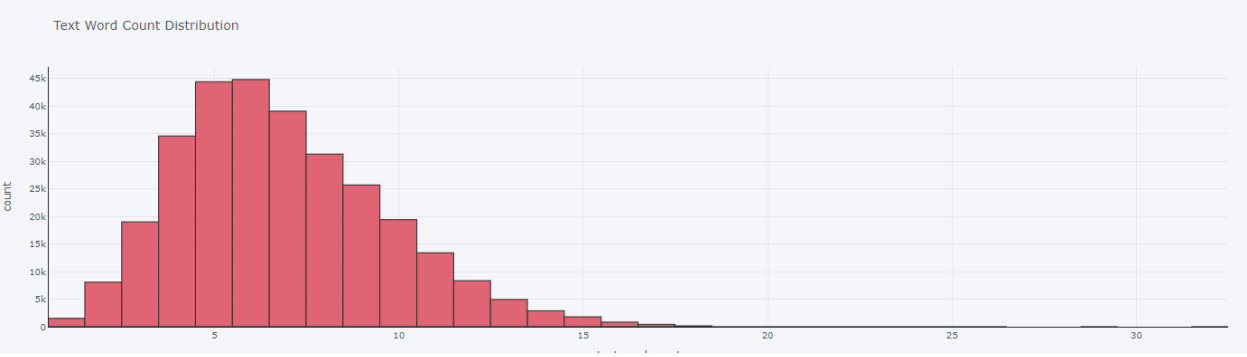


Figure 5.12: Text Word Count Distribution Visualization

The next thing is to compare the training and validation dataset to see the difference between their text_len and text_word_count. The figure 5.14 shows the comparison of text_len between the two datasets and figure 5.16 shows the comparison of text_word_count. Each comparison shows that they have similarities but as the validation dataset is a smaller dataset the pattern is much smaller compared to the training dataset.

```
[19] pal = sns.color_palette()
     train = df['raw_address'].apply(len)
     valid = validation['raw_address'].apply(len)

     plt.figure(figsize=(15, 10))
     plt.hist(train, bins=180, range=[0, 180], color=pal[2], label='train')
     plt.hist(valid, bins=180, range=[0, 180], color=pal[1], alpha=0.5, label='test')
     plt.title('Character Count', fontsize=15)
     plt.legend()
     plt.xlabel('Number of characters', fontsize=15)
     plt.ylabel('Probability', fontsize=15);
```

Figure 5.13: Text_len comparison code

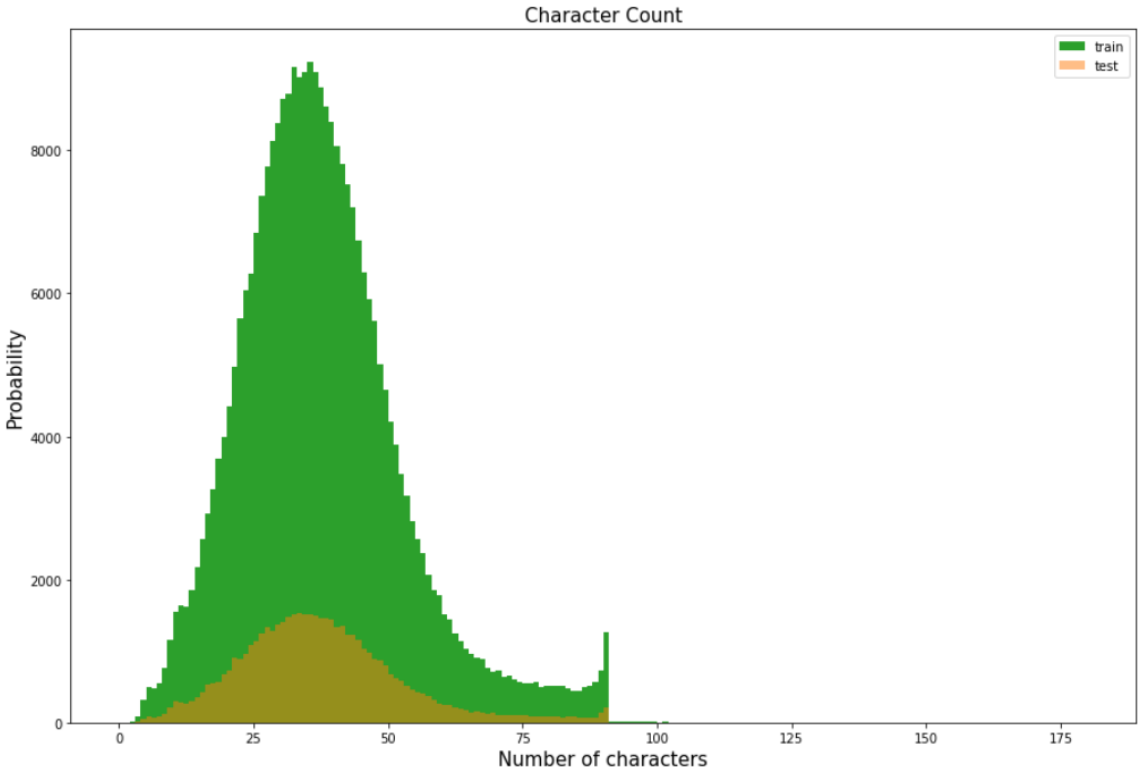


Figure 5.14: Text_len comparison result

The code on Figure 5.13 plot histogram where it shows the text_len on Figure 5.14. The figure 5.14 explains the comparison between the test and train dataset where the green is the train and yellowish is test dataset. Both of the dataset is at the same average length of words where it is around 37 words.

```
[20] train = df['raw_address'].apply(lambda x: len(x.split(' ')))
     valid = validation['raw_address'].apply(lambda x: len(x.split(' ')))

     plt.figure(figsize=(15, 10))
     plt.hist(train, bins=40, range=[0, 40], color=pal[2], label='train')
     plt.hist(valid, bins=40, range=[0, 40], color=pal[1], alpha=0.5, label='valid')
     plt.title('Word Count', fontsize=15)
     plt.legend()
     plt.xlabel('Number of words', fontsize=15)
     plt.ylabel('Probability', fontsize=15);
```

Figure 5.15: Text_word_count comparison code

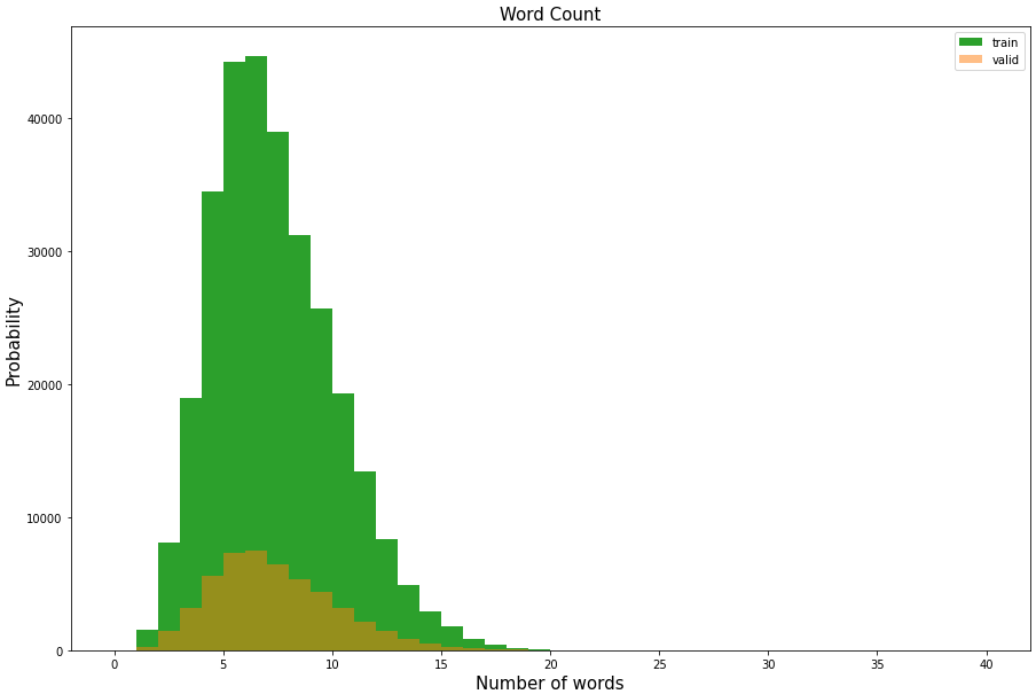


Figure 5.16: Text_word_count comparison result

The figure 5.15 are the code to plot histogram that shows the number of words of the raw address. The plot is shown in Figure 5.16 where the average number of words is around 13 words and the most number of words is 19 words.

5.4. Data Visualization

In this data visualization stage, it will include n-gram distribution visualization, most common words chart, and word cloud to help visualized the data in more interesting perspective. The n-gram allow the developer to see the top combination of words that appear commonly as well as knowing the most used words as well as seeing interesting concept of word cloud where the size of the word means the frequency of it.

5.4.1. Top Unigram Distribution

```
[21] def get_top_n_words(corpus, n=None):
        """
        List the top n words in a vocabulary according to occurrence in a text corpus.
        """
        vec = CountVectorizer().fit(corpus)
        bag_of_words = vec.transform(corpus)
        sum_words = bag_of_words.sum(axis=0)
        words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
        words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
        return words_freq[:n]

[39] enable_plotly_in_cell()
     unigrams = get_top_n_words(df['raw_address'], 20)
     df1 = pd.DataFrame(unigrams, columns = ['Text', 'count'])

     df1.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Unigrams (Raw Address)',orientation='h')
```

Figure 5.17: Top Unigram Code (raw_address)

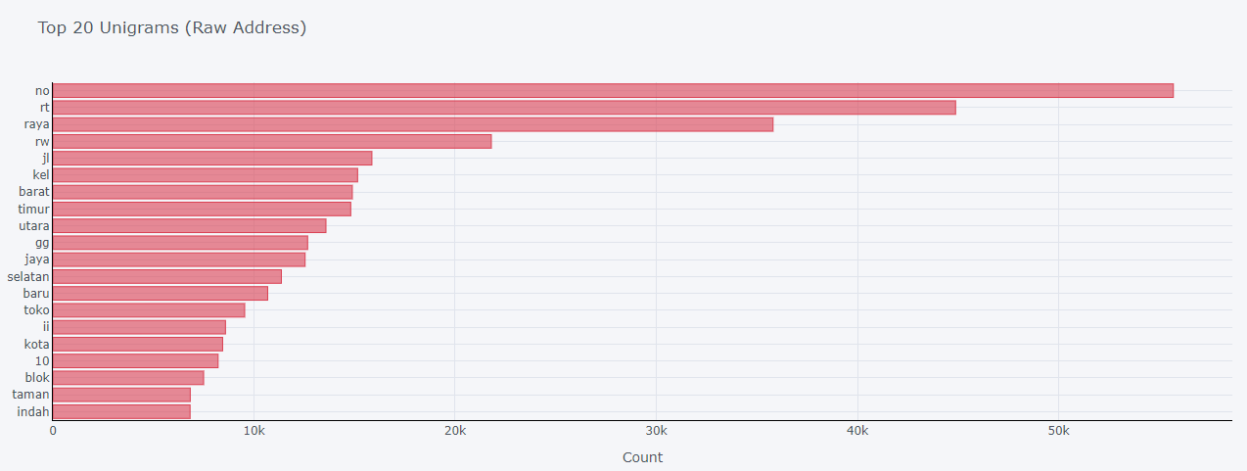


Figure 5.18: Top Unigram Distribution (raw_address)

The top 20 unigram distribution for “raw_address” is shown on Figure 5.18 where “no” which means number and usually used in pointing out the number of unit or building the customer lived in. The second is “rt” is used when stating the community unit, this applies to the fourth which is “rw” which implies the neighborhood unit.

```
[40] enable_plotly_in_cell()
     df_cleaned = df.dropna()
     unigrams = get_top_n_words(df_cleaned['POI'], 20)
     df2 = pd.DataFrame(unigrams, columns = ['Text', 'count'])

     df2.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Unigrams (POI)',orientation='h')
```

Figure 5.19: Top Unigram (POI)

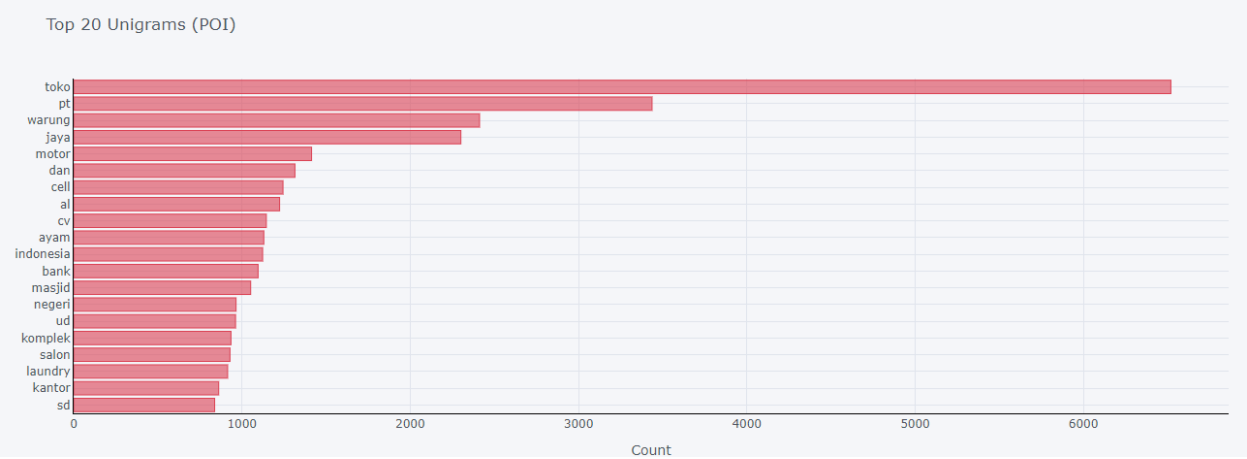


Figure 5.20: Top Unigram Distribution (POI)

For top 20 POI unigram distribution the number one is “toko” which translates to “store” in English, the same applies also to “warung” which is the third. For the second place “pt” it is commonly used for company and the best example to explain this concept would be “sdn bhd” where most Malaysia company ends with “sdn bhd” but as for Indonesia most of the company there starts with “pt”.

```
[41] enable_plotly_in_cell()
     unigrams = get_top_n_words(df_cleaned['street'], 20)
     df2 = pd.DataFrame(unigrams, columns = ['Text', 'count'])

     df2.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Unigrams (street)',orientation='h')
```

Figure 5.21: Top Unigram Code (street)

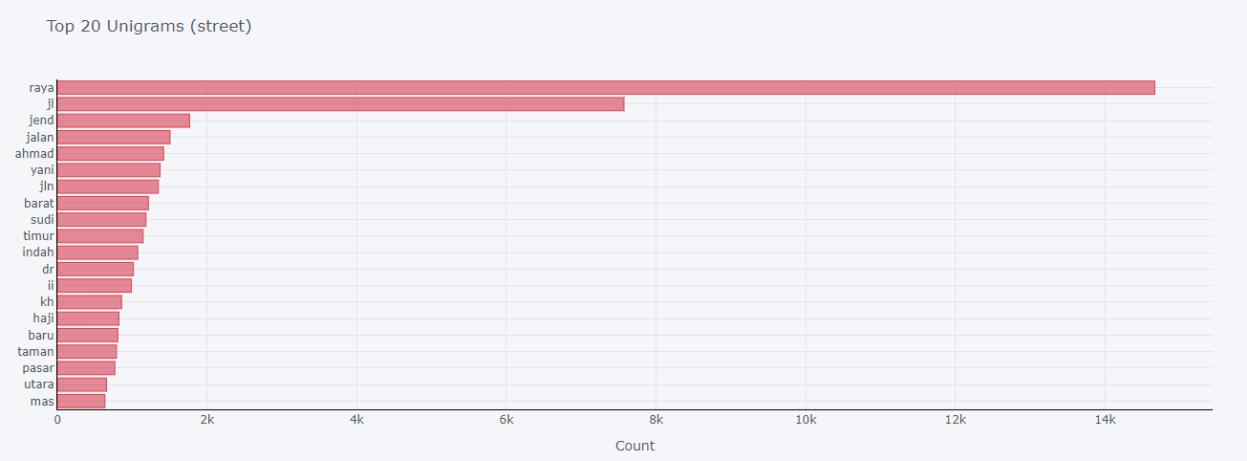


Figure 5.22: Top Unigram Distribution (street)

The figure 5.22 shows that “raya” are at the first place when it comes to street. For the word “raya”, it is commonly used word for main road thus it makes sense as most of the road name are followed with “raya” and as for “Jl” which is the abbreviated version of “jalan” which translated to road itself thus the second place is self-explanatory.

5.4.2. Top Bigram Distribution

```
[42] def get_top_n_gram(corpus,ngram_range,n=None):
         vec = CountVectorizer(ngram_range=ngram_range).fit(corpus)
         bag_of_words = vec.transform(corpus)
         sum_words = bag_of_words.sum(axis=0)
         words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
         words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
         return words_freq[:n]
```

```
[52] enable_plotly_in_cell()
     bigrams = get_top_n_gram(df['raw_address'],(2,2),20)
     df1 = pd.DataFrame(bigrams, columns = ['Text', 'count'])

     df1.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Bigrams (raw_address)',orientation='h')
```

Figure 5.23: Top Bigram Code

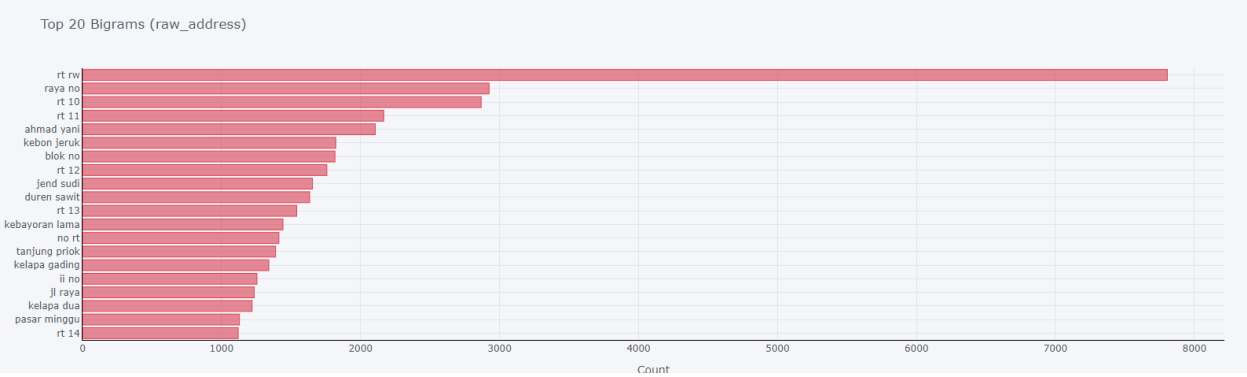


Figure 5.24: Top Bigram Distribution (raw_address)

For the combination of two words which is bigram as shown in Figure 5.24 at the top is “rt rw” which is the combination of Neighborhood and community unit. Sometimes it also followed up with numbers such as shown in the third, fourth, eighth, eleventh, and twentieth. These are all common as the customer from Indonesia tends to enter “rt” or “rw” whenever address is asked to be typed in as they think that It is absolutely necessary to include in for the deliveryman to find their house.

```
[46] enable_plotly_in_cell()
     bigrams = get_top_n_gram(df_cleaned['POI'],(2,2),20)
     df2 = pd.DataFrame(bigrams, columns = ['Text', 'count'])

     df2.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Bigrams (POI)',orientation='h')
```

Figure 5.25: Top Bigram Code (POI)

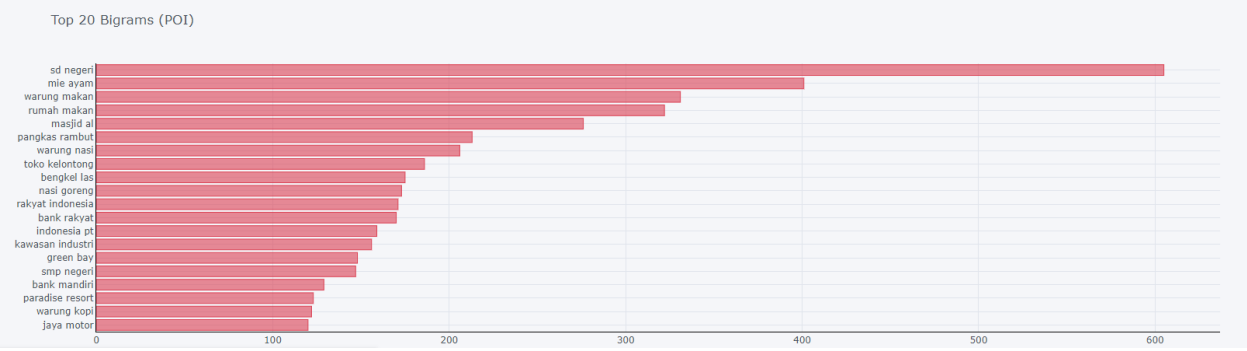


Figure 5.26: Top Bigram Distribution (POI)

For top bigram distribution of POI, it is much more random but at the top is “sd negeri” which is commonly words used for school where “sd” translated to primary school and “negeri” is translated to the country which means it is not a private school.

```
[47] enable_plotly_in_cell()
     bigrams = get_top_n_gram(df_cleaned['street'],(2,2),20)
     df3 = pd.DataFrame(bigrams, columns = ['Text', 'count'])

     df3.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Bigrams (street)',orientation='h')
```

Figure 5.27 Top Bigram Code (street)

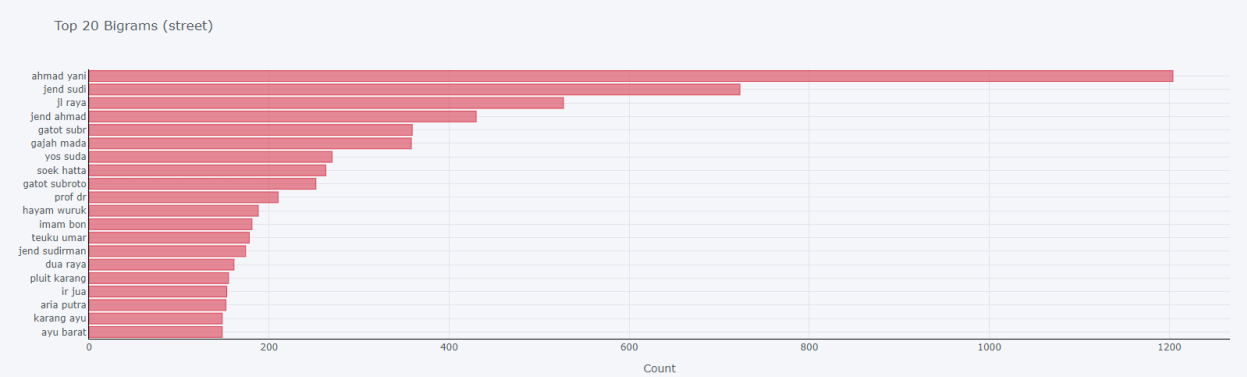


Figure 5.28: Top Bigram Distribution (street)

The figure 5.28 shows that most used word is “ahmad yani” this is commonly used word for street name as the word is a person name and he is a commander of Indonesian army and is regarded as war hero. The same applies to “jend sudi”, “gatot sbr”, “gajah mada”, “soek hatta”, etc. Most of them are name from the respected people of Indonesia which was used as street name to remember them.

5.4.3. Top Trigram Distribution

```
[49] enable_plotly_in_cell()
     trigrams = get_top_n_gram(df['raw_address'],(3,3),20)
     df1 = pd.DataFrame(trigrams, columns = ['Text' , 'count'])

     df1.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Trigrams (raw_address)',orientation='h')
```

Figure 5.29: Top Trigram Code

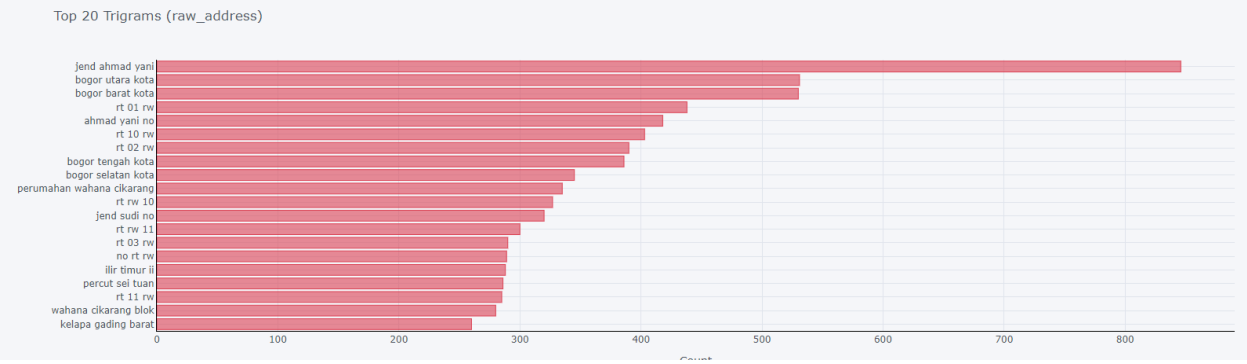


Figure 5.30: Top Trigram Distribution (raw_address)

For the top trigram distribution shown in Figure 5.30 the most commonly used word for raw_address is “jend ahmad yani” also mean the same as “ahmad yani” and “jend sudi” which is shown on figure 5.28. It is a respected person name that is commonly used for road name. As for “bogor utara kota” and “bogor barat kota”, this shows which city it is from for example “bogor” implies the city name, “utara” means north, and “kota” translated to city.

```
[50] enable_plotly_in_cell()
     trigrams = get_top_n_gram(df_cleaned['POI'],(3,3),20)
     df2 = pd.DataFrame(trigrams, columns = ['Text' , 'count'])
     df2.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Trigrams (POI)',orientation='h')
```

Figure 5.31: Top Trigram Distribution (POI)

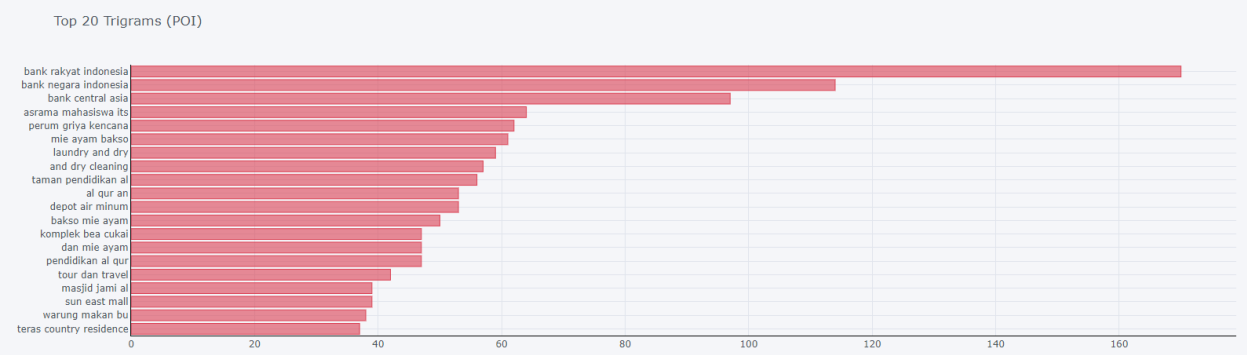


Figure 5.32: Top Trigram Distribution (POI)

For the Top POI distribution of Trigram shown in Figure 5.32 the most common point of interest is bank, it is appeared on top 3 of the chart. Other than that, the other are mostly just random point of interest.

```
[51] enable_plotly_in_cell()
     trigrams = get_top_n_gram(df_cleaned['street'],(3,3),20)
     df3 = pd.DataFrame(trigrams, columns = ['Text' , 'count'])

     df3.groupby('Text').sum()['count'].sort_values(ascending=True).iplot(
         kind='bar', xTitle='Count', linecolor='black',color='red', title='Top 20 Trigrams (street)',orientation='h')
```

Figure 5.33: Top Trigram Code (street)

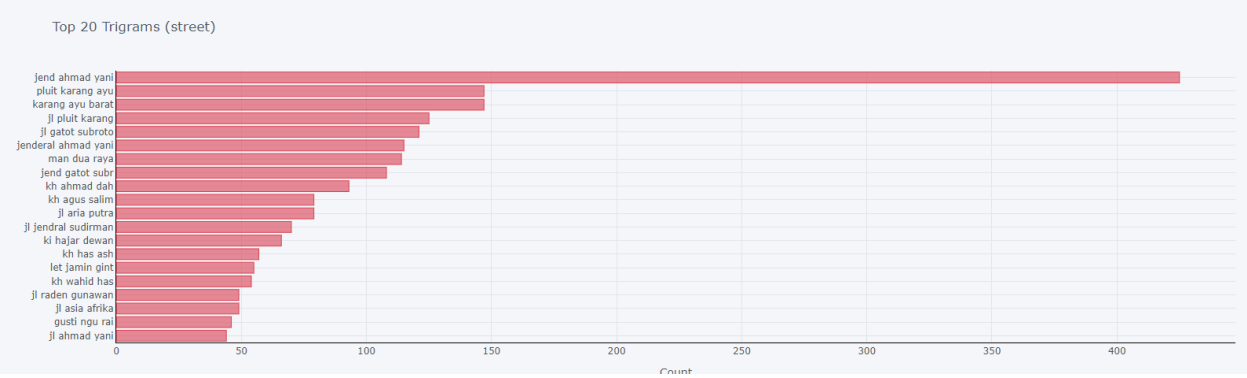


Figure 5.34: Top Trigram Distribution (street)

For the top trigram distribution of street, the most common is also “jend ahmad yani” as the one appeared in Figure 5.30 and the abbreviated version of it in Figure 5.28.

5.4.4. Most Common Words

```
[53] def get_top_n_words(corpus, n=None):
        """
        List the top n words in a vocabulary according to occurrence in a text corpus.
        """
        vec = CountVectorizer().fit(corpus)
        bag_of_words = vec.transform(corpus)
        sum_words = bag_of_words.sum(axis=0)
        words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
        words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
        return words_freq[:n]

[54] top_words = get_top_n_words(df['raw_address'])
     x = [x[0] for x in top_words[:30]]
     y = [x[1] for x in top_words[:30]]

[56] enable_plotly_in_cell()
     fig = go.Figure([go.Bar(x=x, y=y, text=y)])
     fig.update_layout(uniformtext_minsize=8, uniformtext_mode='hide', title_text='Most Common Words')
```

Figure 5.35: Most Common Words code

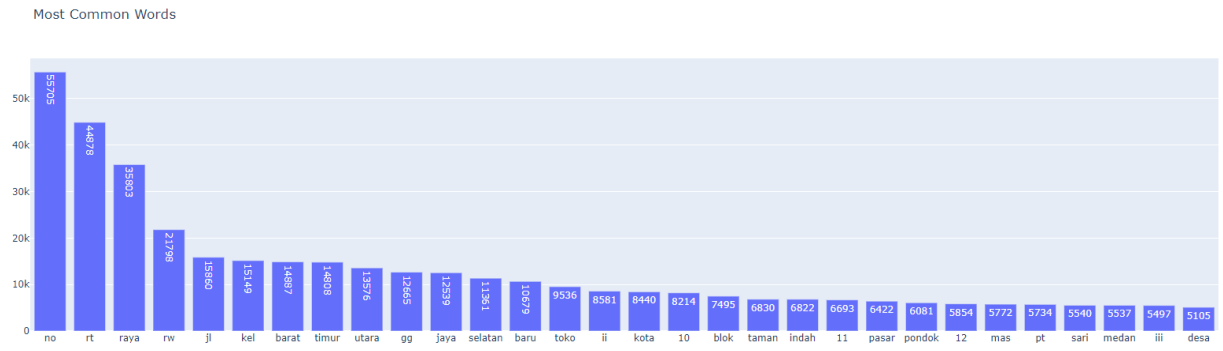


Figure 5.36: Most Common Words Visualization

The most common words used that is shown on Figure 5.36 is “no” which basically means number and it appear for 55705 times throughout the dataset. The second most appeared is “rw” which is the community unit appeared for 44878 times in the dataset.

5.4.5. Word Cloud

```
[58] def plot_cloud(wordcloud):
         plt.figure(figsize=(10, 8)) # Set figure size
         plt.imshow(wordcloud) # Display image
         plt.axis("off"); # No axis details

[59] wordcloud = WordCloud(width = 3000,
                           height = 2000,
                           random_state=1,
                           background_color='black',
                           colormap='Wistia',
                           collocations=False).generate(" ".join(df['raw_address']))

     plot_cloud(wordcloud)
```

Figure 5.37: Word Cloud Code

Figure 5.38: Word Cloud Visualization

From Figure 5.38 the most stood out words would be “raya”, “rw”, “jl”, “kel”, “barat”, “timur”, and “utara”. All of the words mentioned are commonly used words that constantly appear in raw_address and street.

5.5. Data Preprocessing

In this stage after the developer analyzed the missing values, duplicate values, distribution, and the dataset characteristics it was concluded that the data need some cleaning and some preprocessing before it was proceeded to the next phase.

5.5.1 Dataset Splitting

```
[60] import pandas as pd
     pd.options.mode.chained_assignment = None
     df = pd.read_csv('/content/train.csv')

[61] from sklearn.model_selection import train_test_split
     train_df, test_df = train_test_split(df, test_size=0.15, random_state=75)

[63] # train on full dataset
     train_df = pd.read_csv('/content/train.csv')

[64] len(train_df), len(test_df)

     (300000, 45000)
```

Figure 5.39: Test and Train splitting code

Since the test dataset that is collected does not contain the POI/Street variable as shown in Figure 5.1 the developer will not use the test dataset but instead the developer will make his own test dataset from the train.csv that was collected. As it was fairly large dataset which contain 300000 rows, the train data set size is splitted into 15% of the train.csv where the train.csv was used full instead of 85% of it to build the model.

5.5.2 Data Cleaning

```
from string import punctuation
import re

def clean(s):
    res = re.sub(r'(\w)(\()(\w)', '\g<1> \g<2>\g<3>', s)
    res = re.sub(r'(\w)([),.:;]+)(\w)', '\g<1>\g<2> \g<3>', res
    res = re.sub(r'(\w)(\.\()(\w)', '\g<1>. (\g<3>', res)
    res = re.sub(r'\s+', ' ', res)
    res = res.strip()
    return res

def stripclean(arr):
    return [s.strip().strip(punctuation) for s in arr]

def testing(x):
    return [s for s in x]
```

Figure 5.40: Data Cleaning Function

```
[71] train_df['raw_address'] = train_df['raw_address'].apply(lambda x: x.strip())
     train_df['POI'] = train_df['POI/street'].str.split('/').str[0].apply(clean).str.split().apply(stripclean)
     train_df['STR'] = train_df['POI/street'].str.split('/').str[1].apply(clean).str.split().apply(stripclean)
```

Figure 5.41: Data Cleaning application

The data cleaning function was shown in Figure 5.40 to remove all the unnecessary symbols as well as splitting the address into individual word. As shown in Figure 5.41 the function was called which will clean variable and split the address into individual word.

5.5.3 Tokenization and Labelling

```
train_df['tokens'] = train_df['raw_address'].apply(clean).str.split()
train_df['strip_tokens'] = train_df['tokens'].apply(stripclean)
train_df['full_tokens'] = train_df['tokens'].apply(testing)
train_df['labels'] = train_df['tokens'].apply(lambda x: ['O'] * len(x))
train_df['pos_poi'] = train_df['tokens'].apply(lambda x: [-1, -1])
train_df['pos_str'] = train_df['tokens'].apply(lambda x: [-1, -1])
```

Figure 5.42: Token and Tag Labelling code (Train)

```
[67] train_df.head()
```

	id	raw_address	POI/street	POI	STR	tokens	strip_tokens	full_tokens	labels	pos_poi	pos_str
0	0	jl kapuk timur delta sili iii lippo cika 11 a ...	/jl kapuk timur delta sili iii lippo cika	[]	[jl, kapuk, timur, delta, sili, iii, lippo, cika]	[jl, kapuk, timur, delta, sili, iii, lippo, ci...	[jl, kapuk, timur, delta, sili, iii, lippo, ci...	[jl, kapuk, timur, delta, sili, iii, lippo, ci...	[O, O, O, O, O, O, O, O, O, O, O, O]	[-1, -1]	[-1, -1]
1	1	aye, jati sampurna	/	[]	[]	[aye,, jati, sampurna]	[aye, jati, sampurna]	[aye,, jati, sampurna]	[O, O, O]	[-1, -1]	[-1, -1]
2	2	setu siung 119 rt 5 1 13880 cipayung	/siung	[]	[siung]	[setu, siung, 119, rt, 5, 1, 13880, cipayung]	[setu, siung, 119, rt, 5, 1, 13880, cipayung]	[setu, siung, 119, rt, 5, 1, 13880, cipayung]	[O, O, O, O, O, O, O, O]	[-1, -1]	[-1, -1]
3	3	toko dita, kertosono	toko dita/	[toko, dita]	[]	[toko, dita,, kertosono]	[toko, dita, kertosono]	[toko, dita,, kertosono]	[O, O, O]	[-1, -1]	[-1, -1]
4	4	jl. orde baru	/jl. orde baru	[]	[jl. orde, baru]	[jl., orde, baru]	[jl, orde, baru]	[jl., orde, baru]	[O, O, O]	[-1, -1]	[-1, -1]

Figure 5.43: Token and Tag Labelling result (Train)

```
test_df['raw_address'] = test_df['raw_address'].apply(lambda x: x.strip())
test_df['tokens'] = test_df['raw_address'].apply(clean).str.split()
```

Figure 5.44: Token and Tag Labelling code (Test)

```
[69] test_df.head()
```

	id	raw_address	POI/street	tokens
90142	90142	lom 88 asrikaton	/	[lom, 88, asrikaton]
163531	163531	varia usaha ungaran, peri kem pudakpayung	/	[varia, usaha, ungaran,, peri, kem, pudakpayung]
233950	233950	hutan gar no 7 20371 percut sei tuan	/gar	[hutan, gar, no, 7, 20371, percut, sei, tuan]
126157	126157	wardah gor srik ton,	wardah gorden/srik ton	[wardah, gor, srik, ton,]
96808	96808	green puri 7 cengkareng	/green puri 7	[green, puri, 7, cengkareng]

Figure 5.45: Token and Tag Labelling result (Test)

As shown in Figure 5.43 and 5.45 the dataset is tokenized meaning that the words are turned into tokens to passed for Name Recognition Task (NER) in the model buildings phase.

5.5.4 Building Word List & Token Labelling

To complete the abbreviation of the words that exists in raw_address the developer will build a dictionary word list to complete the words. Additionally, the tokens that were created will be labelled, the label will be categorized into different parts where it was sorted from the most useful data to the least useful. According to Alshammari, & Alanazi (2021) NER task used different schemes such as following:

- IO: is the simplest scheme that can be applied to this task. In this scheme, each token from the dataset is assigned one of two tags: an inside tag (I) and an outside tag (O). The I tag is for named entities, whereas the O tag is for normal words. This scheme has a limitation, as it cannot correctly encode consecutive entities of the same type.

- IOB: This scheme is also referred to in the literature as BIO and has been adopted by the Conference on Computational Natural Language Learning (CoNLL) [1]. It assigns a tag to each word in the text, determining whether it is the beginning (B) of a known named entity, inside (I) it, or outside (O) of any known named entities.

- IOE: This scheme works nearly identically to IOB, but it indicates the end of the entity (E tag) instead of its beginning.

- IOBES: An alternative to the IOB scheme is IOBES, which increases the amount of information related to the boundaries of named entities. In addition to tagging words at the beginning (B), inside (I), end (E), and outside (O) of a named entity. It also labels single-token entities with the tag S.

- BI: This scheme tags entities in a similar method to IOB. Additionally, it labels the beginning of non-entity words with the tag B-O and the rest as I-O.

- IE: This scheme works exactly like IOE with the distinction that it labels the end of non-entity words with the tag E-O and the rest as I-O.

- BIES: This scheme encodes the entities similar to IOBES. In addition, it also encodes the non-entity words using the same method. It uses B-O to tag the beginning of non-entity words, I-O to tag the inside of non-entity words, and S-O for single non-entity tokens that exist between two entities.

With the consideration of the complexity of raw_address instead of using simple scheme such as IO, the best scheme for this scenario would be IOBES scheme as it allows more information. While using the IOBES scheme an additional tag will be included to label the incomplete words that needed to be fixed which is later be passed to the word list that are built.

5.6. Model Buildings

5.6.1. Introduction

Fastai is a library focused on machine learning written on Python. It is of the most widely used deep learning frameworks. It allows us to train more accurate models faster, with less data, and in less time and money (Jwalapuram, 2021). Jeremy Howard the developer of Fastai mentions, everything's much easier with Fastai thanks to less codes written by developer meaning it provides flexibility, speed and also ease-of-use at the identical time. It offers an excellent deal of features additionally as functionality that produces developers customize the high-level API without getting attached low-level API parts. One instance of this customization is DataBlock, which helps you to load the info in a very detailed way.

The BERT model from Fastai works based on transformers library. It is a library that can provide thousands of pretrained model which can be applied for unstructured text and is what the developer needed (PyPI, 2022). By incorporating the library with Fastai it allows to increase the learning rate as well as providing gradual unfreezing. According to Roberti (2019) two authors which are Keita Kurita and Dev Sharma have testified that pytorch_transformers and pytorch_pretrained_bert are compatible with fastai. The main structure of the model contain three major variable, the first is the model class which are used to keep the pre-train model as well as to load the model itself. The second is the tokenizer class which is used to preprocess the elements to make it passable for BERT model and lastly is the configuration class which is to configurate the model that is to be store or loaded.

```
from transformers import BertForSequenceClassification, BertTokenizer, BertConfig
from transformers import RobertaForSequenceClassification, RobertaTokenizer, RobertaConfig
from transformers import XLNetForSequenceClassification, XLNetTokenizer, XLNetConfig
from transformers import XLMForSequenceClassification, XLMTokenizer, XLMConfig
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer, DistilBertConfig

MODEL_CLASSES = {
    'bert': (BertForSequenceClassification, BertTokenizer, BertConfig),
    'xlnet': (XLNetForSequenceClassification, XLNetTokenizer, XLNetConfig),
    'xlm': (XLMForSequenceClassification, XLMTokenizer, XLMConfig),
    'roberta': (RobertaForSequenceClassification, RobertaTokenizer, RobertaConfig),
    'distilbert': (DistilBertForSequenceClassification, DistilBertTokenizer, DistilBertConfig)}

model_type = 'roberta'

model_class, tokenizer_class, config_class = MODEL_CLASSES[model_type]
```

Figure 5.46: Example of Loading and Storing Model Types

5.6.2. Tokenization

To get the dataset ready for the BERT model the developer needs to process the data first which are called tokenization as the pre-trained model that the developer will fine-tune needed the data to be passed first through tokenization. One of the methods to tokenize the unstructured text is by using tokenizer class from the transformers library.

```
processor = [TokenizeProcessor(tokenizer=tokenizer,…),
```

Figure 5.47: Example of Tokenizing

However, with the complexity of the unstructured text the developer will build a custom tokenizer which are more complex than the code shown in Figure 5.47. There are three objects that need understanding to build custom tokenizer class, This example is shown on Figure 5.48.

```
class TransformersBaseTokenizer(BaseTokenizer):
    """Wrapper around PreTrainedTokenizer to be compatible with fast.ai"""
    def __init__(self, pretrained_tokenizer: PreTrainedTokenizer, model_type = 'bert', **kwargs):
        self._pretrained_tokenizer = pretrained_tokenizer
        self.max_seq_len = pretrained_tokenizer.max_len
        self.model_type = model_type

    def __call__(self, *args, **kwargs):
        return self

    def tokenizer(self, t:str) -> List[str]:
        """Limits the maximum sequence length and add the spesial tokens"""
        CLS = self._pretrained_tokenizer.cls_token
        SEP = self._pretrained_tokenizer.sep_token
        if self.model_type in ['roberta']:
            tokens = self._pretrained_tokenizer.tokenize(t, add_prefix_space=True)[:self.max_seq_len - 2]
            tokens = [CLS] + tokens + [SEP]
        else:
            tokens = self._pretrained_tokenizer.tokenize(t)[:self.max_seq_len - 2]
            if self.model_type in ['xlnet']:
                tokens = tokens + [SEP] +  [CLS]
            else:
                tokens = [CLS] + tokens + [SEP]
        return tokens

transformer_tokenizer = tokenizer_class.from_pretrained(model_name)
transformer_base_tokenizer = TransformersBaseTokenizer(pretrained_tokenizer = transformer_tokenizer, model_type = model_type)
fastai_tokenizer = Tokenizer(tok_func = transformer_base_tokenizer, pre_rules=[], post_rules=[])
```

Figure 5.48: Custom Tokenizer Example

There's three object. TokenizeProcessor, Tokenizer, and BaseTokenizer. The TokenizeProcessor take 'tokenizer' as Tokenizer object where it will take it as 'tok_func' argument as the BaseTokenizer object which will then be impleneted to 'tokenizer(t:str) -> List[str]' which receives 't' as text and return the token into the list that was declared.

BERT: [CLS] + tokens + [SEP] + padding

The formula above are from Hugging Face Documentation where it is one of the five models that is described in it. Though on Figure 5.48 there's no padding included as fastai automatically did it during the creation of datablock object, a collection of DataLoaders when the databunch function was called (Gilliam, 2019).

5.6.3. Processor

After the custom tokenizer object was build we need to create a custom processor where it will pass the options over as Fastai added its own tokens by default which will prevent the model to be built, the tokens that needed to be sorted is the [CLS] and [SEP] interfered by Fastai from the custom tokenizer shown in Figure 5.48.

```
transformer_vocab =  TransformersVocab(tokenizer = transformer_tokenizer)
numericalize_processor = NumericalizeProcessor(vocab=transformer_vocab)

tokenize_processor = TokenizeProcessor(tokenizer=fastai_tokenizer,
                                       include_bos=False,
                                       include_eos=False)

transformer_processor = [tokenize_processor, numericalize_processor]
```

Figure 5.49: Custom Processor example

5.6.4. Databunch

From the Hugging Face documentation, the BERT model have absolute positon embeddings where the developer need to follow thus the input need to be positioned in the correct order thus the datablock are used for. To build a proper datablock it need to be based on Figure 5.49 the custom processor that is built.

```
pad_first = bool(model_type in ['xlnet'])
pad_idx = transformer_tokenizer.pad_token_id

databunch = (TextList.from_df(train, cols='Phrase', processor=transformer_processor)
             .split_by_rand_pct(0.1,seed=seed)
             .label_from_df(cols= 'Sentiment')
             .add_test(test)
             .databunch(bs=bs, pad_first=pad_first, pad_idx=pad_idx))
```

Figure 5.50: Datablock example

5.6.5. Training

The fastai build-in features can now finally be used to train models where ULMFiT method will be used where Slanted Triangular Learning Rates, Discriminate Learning Rate and gradually unfreeze the model. Thus, the first step is to freeze the group of classifier with code shown on Figure 5.51. As for Slanted Triangular Learning Rates the function ‘fit_one_cycle will be used to find the optimum learning rate utilizing also ‘lr_find’. For example, Figure 5.52 shows that the value at the slight minimum before the loss improves is the best value which is 2×10^{-3}. This step will be repeated until all of the group is unfrozen.

```
learner.freeze_to(-1)
```

Figure 5.51: Freezing classifer

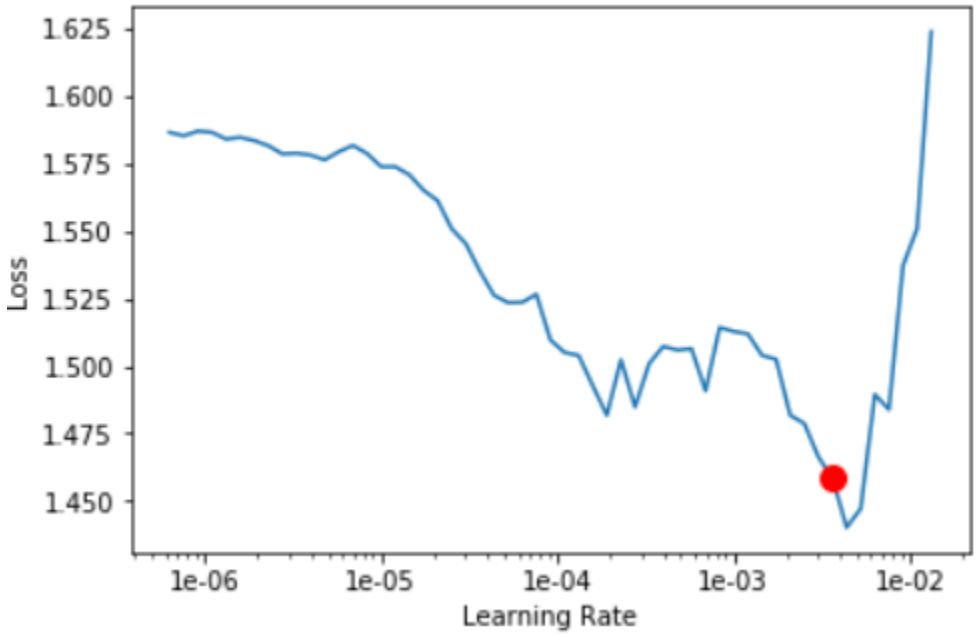


Figure 5.52: Learning Rate & Loss graph

5.7. Summary

It was explained how the transformer library are to be combined with Fastai library where it is able to increase the efficiency and to train a better model with some modifications and implementation of codes. It also allows the used of slanted triangular learning rate, discriminate learning rate as well as gradual unfreezing which allow the developer to generate result without properly fine-tune the parameters.

CHAPTER 6: RESULTS AND DISCUSSION

6.1. Introduction

In this section of the report, the author will discuss the method of evaluation as well as the deployment of the model by using the flask framework to deploy a simple one-page website that's able receive input and passing the input to the built model and printing the extraction of the element.

6.2. Model Evaluation

The address training dataset contain 300000 rows of raw address inputted by customers. It also includes the 'POI/Street' which are the element of target. The number of epochs for the training was set at 20 but the run time for the model only allows 4 hours thus the epoch that was created was only 15 epochs.

Table 6.1: Precision, Recall & F1 Score for 15 Epochs

Number	Precision	Recall	F1
1	0.771	0.882	0.821
2	0.781	0.898	0.834
3	0.789	0.902	0.841
4	0.792	0.900	0.841
5	0.791	0.901	0.841
6	0.792	0.889	0.837
7	0.794	0.884	0.835
8	0.793	0.873	0.829
9	0.792	0.870	0.828
10	0.787	0.849	0.815
11	0.794	0.855	0.822

12	0.791	0.849	0.817
13	0.793	0.851	0.819
14	0.792	0.848	0.817
15	0.792	0.851	0.819
Average	0.7896	0.873466667	0.827733333

Overall, the selection of the model will be depending on the accuracy score of the model itself where it is measured by the number of correct predictions. The following is the accuracy metric.

$$accuracy((p_i, s_i), (\hat{p}_i, \hat{s}_i)) = \begin{cases} 1 \ if \ p_i == \hat{p}_i \ and \ s_i == \hat{s}_i \\ 0 \ otherwise \end{cases}$$

Where:

p_i = the actual POI name for ith address
$\hat{p}_i$ = the predicted POI name for ith address
s_i = the actual street name for ith address
$\hat{s}_i$ = the predicted street name for ith address

The addresses are often followed by missing POI or street elements thus for such situations, the specific element should be left empty. The following formula is to calculate the average accuracy score of the model. Following Table 6.2 where the author model's accuracy are shown.

$$score = \frac{1}{n} \sum_{i=1}^{n} (accuracy((p_i, \hat{p}_i), (s_i, \hat{s}_i)))$$

Where:

n = the total number of addresses
$accuracy$ = the function provided above

Table 6.2: Accuracy for the prediction models

No	Train Score	Validation Score	AVG
1	0.7035	0.5966	0.6500
2	0.7766	0.6535	0.7150
3	0.7800	0.6555	0.7177
4	0.7194	0.6610	0.6902
AVG	0.7448	0.6416	0.6932

Table 6.3: Accuracy for Existing or Similar element extraction models

No	Dataset	Source	Accuracy
1	RCV1-v2, Divorce, Loan, Labor Dataset	Chen, Zhang, Ye & Li (2021)	87.55%
2	Korean Building Regulation Sentences	Song, Lee, Choi & Kim (2020)	83%
3	Cancer Dataset	Si & Roberts (2018)	94.52%
4	Access Control Policy (ACP) Dataset	Alohaly, Takabi & Blanco (2018)	86.3%
5	Kawasaki Disease Dataset	Kuo, Rao, Maeharak, Doan, Chaparro, Day & Hsu, (2016)	79%
6	Electronic Medical Dataset	Sun, Cai, Li, Liu, Fang, & Wang (2018)	78.8%
7	Research Articles Sentences	Salloum, Al-Emran, Monem & Shaalan (2018)	86.64%
Average			85.12%

In comparison of similar model from existing studies where it has average of 85% accuracy whereas the model built by the author on Table 6.2 only has 69.3% accuracy. This means that the model could be improved as the model and system that is shown on Table 6.3 are all element extraction model similar to the current model built by author but for Indonesia raw address.

6.3. Model Deployment

The deployment of the prototype model is in the form of website application by using the Flask framework. The main reason of using the prototype model is because it takes less time to load but still show how the base model works. The prototype model that is used does not include grammar correction but does include NER model with 61% accuracy. The deployment will be using PyCharm IDE as the web application runs on py file and not ipynb file.

6.3.1. Website Design

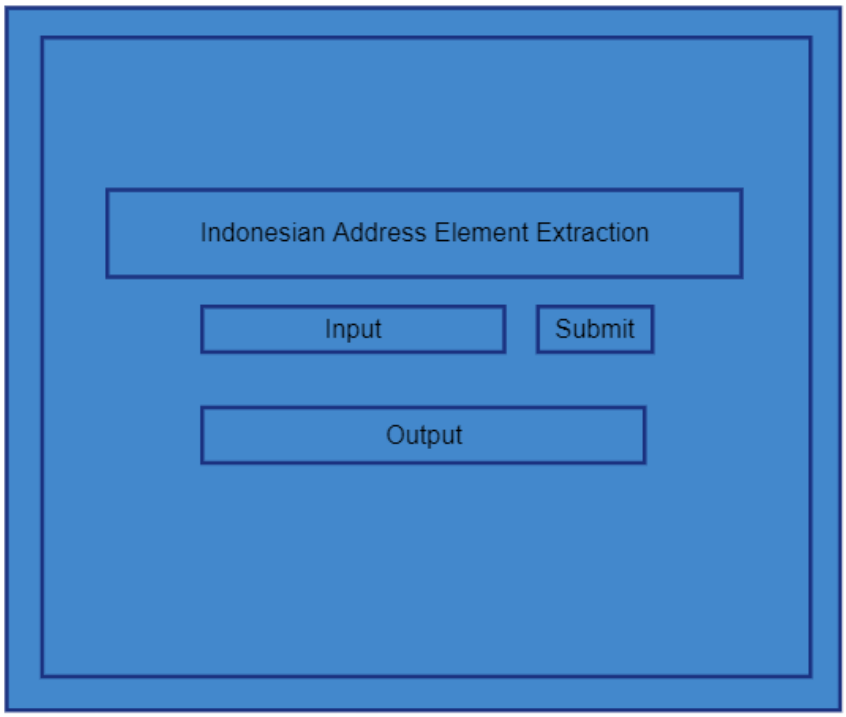


Figure 6.1: Interface Mock-up design

The mock-up design of the website application is shown on Figure 6.1 where there will be a title, an input box to enter the raw address, submit box where once the address is typed in the submit button is there to pass the raw address to the model and the output box will display the elements that were extracted from the raw address.

```html
<!DOCTYPE html>
<html >
<!--From https://codepen.io/frytyler/pen/EGdtg-->
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
    <div class="background">
        <div class="login">
            <h1>Indonesian Address Element Extraction</h1>
                <!-- Receive input and pass to the model -->
                <form action="{{ url_for('predict')}}"method="post">
                    <input type="text" name="raw_address" placeholder="Your address" required="required" />
                    <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
                </form>
            <br>
            <br>
            {{ poi }}
            <br>
            <br>
            {{ srt }}
        </div>
    </div>
</body>
</html>
```

Figure 6.2: HTML code for the website design

For the main page of the deployment, the code is shown in Figure 6.2 where it covers the base design shown in Figure 6.1. It includes the title, input, and submit button. As for the output, it will be passed through variable and printed as shown in Figure 6.2 where there's {{ poi }} and {{ srt }}. With CSS the website design is shown on Figure 6.3.

Figure 6.3: Website Application Design

Figure 6.4: Website Application Design with output

6.3.2. Deployment with Flask

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import spacy
```

Figure 6.5: Library Imports for Flask deployment

The libraries that needed to be imported when deploying the website application. The spacy module is imported to load the model.

```
@app.route('/')
def home():
    return render_template('index.html')
```

Figure 6.6: Homepage html template render

The code shown in Figure 6.6 is to create the main page of the website application running the index.html file which is shown in Figure 6.2.

```python
@app.route('/predict',methods=['POST'])
def predict():
    float_features = [str(x) for x in request.form.values()]
    doc = nlp(float_features[0])

    results = {}

    elements = [(X.text, X.label_) for X in doc.ents]
    poi_elements = [x for x in elements if x[1] == 'POI']
    srt_elements = [x for x in elements if x[1] == 'SRT']

    if not poi_elements:
        poi_elements = [('', '')]

    if not srt_elements:
        srt_elements = [('', '')]

    results[id] = (poi_elements, srt_elements)


    return render_template("index.html", poi = "Point of Interest: " + results[id][0][0][0],
                           srt = "Street: " + results[id][1][0][0])
```

Figure 6.7: Passing the variable to display output (predict page)

For the prediction page, the code includes calling requesting input from the user and passing the input as string to the model where later will be passed as an array and the elements extracted. In the return statement the ‘poi’ and ‘srt’ will be passed to index.html where later will be displayed as output.

6.4. Results and Discussion

The models created by the author can be improved as based on existing systems and models they have achieved accuracy higher than 80% whereas the model created by author only barely achieve above 60% accuracy. As for the web deployment, it has the main page where it allows users to input the raw address and after the submit button was pressed, output will be delivered displaying the elements that is extracted from the raw address.

CHAPTER 7: CONCLUSIONS AND REFLECTIONS

7.1. Conclusions

By being able to extract and predict the address of customers, the company can lower the costs of labor and time and improve the efficiency of the operation. Especially since most online retail marketing required the customer to enter manually their address which may be incorrect or incomplete. The models that are built progress step by step following the methodology of CRISP-DM. Based on the research, most of the extraction is done through the tokenization classification method, this shows how useful is it to tokenize the text as without tokenization most models cannot process the model. Though Google Colab is totally free and premium, it should be noted that the fact security is not that safe thus if the project involves a large-scale company, it is much more suggested to not rely on Google Colab. The CRISP-DM method is a viable methodology but only because this project involves E-commerce which means there needs to be a phase made to understand the business itself. SEMMA methodology is preferable if the project only includes Statistical Analysis or uses SAS Enterprise Miner.

In the data analysis chapter, the author discusses the process of data collection as well as the data contents, how the data is structured normally, the translation and explanation. The data is also explored where the author will able to see the missing percentage, duplicate amount, most commonly used words, and the distribution of the dataset in term of n-gram. Data cleaning are also conducted in the process as well as Tokenization and labelling. The model building technique are also discussed in detail where there will be tokenization, data stacking, and training. After the model was build, evaluation was conducted to evaluate the model. The model was evaluated based on the accuracy metrics and was compared with existing studies where similar models and system was deployed. The model was also deployed using Flask framework as website application with simple design displaying output and allowing users to enter input to be passed to the model.

7.2. Reflections

I have learned so much from writing the investigation report, especially in terms of text-mining knowledge. I did not know a lot of the knowledge listed in the literature review until I did this investigation report. This report has given me a better understanding of text-mining, unstructured data, and the current situation of E-commerce. I honestly think that this project will improve a lot of the E-commerce industry in the world if the system is implemented properly. If more time was given for research and learning, the model could definitely be improved as there are various machine learning modules that can achieve higher accuracy but due to the lack of users and tutorial are provided on the internet, I am not able to fully utilize the modules. The website could be improved by deploying the improved model and including more features to the website by connecting to database to store user inputs to feed the model in order to improve the model.

7.3. Future Works

For my future works, based on the report that was created, I should be familiarizing myself with Phyton and Jupyter Notebook's IDE, learning to use more built-in and open-source module to build better and efficient machine learning. To learn as much as I could to produce better machine learning algorithm that was proposed in this report. Also learning how to utilize flask better to build better website application. In future I would also like to improve the machine learning model by integrating a self-learning machine learning model where it gathers information from user. I would also like to utilize the machine learning that I made to collaborate with Google Map API where it allows the searching of address with Google Map and pinpoint the location from the API. With the help of the report, much research was conducted thus some visualization of the action that needed to be done did come to mind.

REFERENCES

ActiveState. (2022, July 12). What is Matplotlib in python? how to use it for plotting? ActiveState. Retrieved July 17, 2022, from https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

Agrawal, S. (2021). *How to split data into three sets (train, validation, and test) and why?* Medium. Retrieved March 8, 2022, from https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c

Ahmed, F., William De Luca, E. & Nürnberger, A., 2009. Revised N-Gram based Automatic Spelling Correction Tool to Improve Retrieval Effectiveness. *Polibits*, pp.39-48.

Alfonso, V., Boar, C., Frost, J., Gambacorta, L., & Liu, J. (2021). E-commerce in the pandemic and beyond. *BIS Bulletin*, *36*(9).

Alohaly, M., Takabi, H., & Blanco, E. (2018, June). A deep learning approach for extracting attributes of ABAC policies. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies (pp. 137-148).

Alshammari, N., & Alanazi, S. (2021). The impact of using different annotation schemes on named entity recognition. Egyptian Informatics Journal, 22(3), 295-302.

Amplayo, R. K., Lim, S., & Hwang, S. W. (2019, October). Text length adaptation in sentiment classification. In Asian Conference on Machine Learning (pp. 646-661). PMLR.

Antons, D., Grünwald, E., Cichy, P., & Salge, T. O. (2020). The application of text mining methods in innovation research: current state, evolution patterns, and development priorities. *R&D Management*, *50*(3), 329-351.

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational linguistics*, *18*(4), 467-480.

Buzainu, G., 2021. *Delivery Delay | What Causes Late Shipping Delivery? | Eurosender.com*. [online] Eurosender.com. Available at: <https://www.eurosender.com/blog/en/delayed-delivery/> [Accessed 4 May 2021].

Chakravarty, S., 2018. *What is geocoding and how can it help sell products*. [online] Geospatial World. Available at: <https://www.geospatialworld.net/blogs/what-is-geocoding-and-how-can-it-help-sell-products/#:~:text=Geocoding%20is%20the%20process%20where,of%20a%20place%20or%20address.> [Accessed 4 May 2021].

Charniak, E., 2021. Review of "Statistical language learning" by Eugene Charniak. *The MIT Press*, 21(1), pp.104-111.

Chen, Z., Zhang, H., Ye, L., & Li, S. (2021). An Approach Based on Multilevel Convolution for Sentence-Level Element Extraction of Legal Text. Wireless Communications and Mobile Computing, 2021.

Choi, S., Park, G., & Kim, H. W. (2019). A Text Mining Approach to the Analysis of BTS Fever.

Dalianis, H. (2018). *Clinical text mining: Secondary use of electronic patient records*. Springer Nature.

Donges, N., (2020). *A Guide to RNN: Understanding Recurrent Neural Networks and LSTM*. [online] Built In. Available at: <https://builtin.com/data-science/recurrent-neural-networks-and-lstm> [Accessed 4 May 2021].

Ebrahimi, R. (2020). Introduction to fastai(part1). Medium. Retrieved June 13, 2022, from https://medium.com/@rojinebrahimi/introduction-to-fastai-part1-c4c28d53aa9

Escursell, S., Llorach-Massana, P., & Roncero, M. B. (2021). Sustainability in e-commerce packaging: A review. *Journal of cleaner production*, *280*, 124314.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27-34.

Fazal-e-Hasan, S. M., Ahmadi, H., Mortimer, G., Grimmer, M., & Kelly, L. (2018). Examining the role of consumer hope in explaining the impact of perceived brand value on customer–brand relationship outcomes in an online retailing environment. *Journal of Retailing and Consumer Services*, *41*, 101-111.

Fazel, S. Ali. (2021). Re: Can someone recommend what is the best percent of divided the training data and testing data in neural network 75:25 or 80:20 or 90:10? Retrieved from:

https://www.researchgate.net/post/can_someone_recommend_what_is_the_best_percent_of_divided_the_training_data_and_testing_data_in_neural_network_7525_or_8020_or_9010/61c0a90a5d28e9694c77fd30/citation/download.

Fiducia, A. (2022). *Python versus R for Data Analytics.* Developer.com. Retrieved March 8 From: https://www.developer.com/languages/python-vs-r/

Gupta, S. & Nishu, K., (2020). Mapping Local News Coverage: Precise location extraction in textual news content using fine-tuned BERT based language model. *Association for Computational Linguistics*, 1(17), pp.155-162.

Gilliam, W. (2019). Finding data block nirvana (a journey through the FASTAI data block API). Medium. Retrieved June 14, 2022, from https://medium.com/@wgilliam/finding-data-block-nirvana-a-journey-through-the-fastai-data-block-api-c38210537fe4#:~:text=DataBunch%3A,and%20optionally%20test%20LabelList%20instances.

Hotz, N. (2022). What is CRISP DM? Data Science Process Alliance. Retrieved July 17, 2022, from https://www.datascience-pm.com/crisp-dm-2/

Horev, R., (2018). *BERT Explained: State of the art language model for NLP*. [online] Medium. Available at: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270#:~:text=How%20BERT%20works,%2Dwords)%20in%20a%20text.&text=As%20opposed%20to%20directional%20models,sequence%20of%20words%20at%20once.> [Accessed 4 May 2021].

IBM. (2021) *Python vs. R: What's the Difference?* IBM. Retrieved March 8 From: https://www.ibm.com/cloud/blog/python-vs-r

Idreos, S., Papaemmanouil, O., & Chaudhuri, S. (2015). Overview of data exploration techniques. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 277-281).

Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., & Ermon, S. (2016). Combining satellite imagery and machine learning to predict poverty. *Science*, *353*(6301), 790-794.

Johnson, D. (2022). *R Vs Python: What's the Difference?*. Guru99. Retrieved March 8 From: https://www.guru99.com/r-vs-python.html

JournalDev (2021) *Python String Module.* Retrieved March 8 From: https://www.journaldev.com/23788/python-string-module

Jwalapuram, N. (2021). Fast.ai: Training deep learning models with Fast.ai. Analytics Vidhya. Retrieved June 11, 2022, from https://www.analyticsvidhya.com/blog/2021/05/training-state-of-the-art-deep-learning-models-with-fast-ai/

Kaggle.com. (2021). *Shopee Code League - Address Elements Extraction | Kaggle*. From: https://www.kaggle.com/c/scl-2021-ds.

Kuo, T. T., Rao, P., Maehara, C., Doan, S., Chaparro, J. D., Day, M. E., ... & Hsu, C. N. (2016). Ensembles of NLP tools for data element extraction from clinical notes. In AMIA Annual Symposium Proceedings (Vol. 2016, p. 1880). American Medical Informatics Association.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234-1240.

Li, F., Jin, Y., Liu, W., Rawat, B. P. S., Cai, P., & Yu, H. (2019). Fine-tuning bidirectional encoder representations from transformers (BERT)–based models on large-scale electronic health record notes: an empirical study. *JMIR medical informatics*, *7*(3), e14830.

Li, P., Luo, A., Liu, J., Wang, Y., Zhu, J., Deng, Y. & Zhang, J., 2020. Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation. *ISPRS International Journal of Geo-Information*, 9(11), p.635

Lindell, Y., & Pinkas, B. (2002). Privacy preserving data mining. *Journal of cryptology*, *15*(3).

Liu, C. J., Huang, T. S., Ho, P. T., Huang, J. C., & Hsieh, C. T. (2020). Machine learning-based e-commerce platform repurchase customer prediction model. *Plos one*, *15*(12), e0243105

Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. *International Journal of Computer Science and Network Security*, *8*(2), 339-344.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010, September). Recurrent neural network-based language model. In *Interspeech* (Vol. 2, No. 3, pp. 1045-1048).

Mishra, M. (2020, September 2). Convolutional Neural Networks, explained. Medium. Retrieved July 17, 2022, from https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

Moloshnikov, I. A., Sboev, A. G., Rybka, R. B., & Gydovskikh, D. V. (2015). An algorithm of finding thematically similar documents with creating context-semantic graph based on probabilistic-entropy approach. *Procedia Computer Science*, *66*, 297-306.

Moore, K., Chumbley, A., & Khim, J. (2022). Context free grammars. Brilliant Math & Science Wiki. Retrieved July 17, 2022, from https://brilliant.org/wiki/context-free-grammars/#:~:text=A%20context-free%20grammar%20is,%2C%20compiler%20design%2C%20and%20linguistics.

Navone, F. C. (2020) *Python Read JSON File – How to Load JSON from a File and Parse Dumps*. Free Code Camp. Retrieved March 8 From: https://www.freecodecamp.org/news/python-read-json-file-how-to-load-json-from-a-file-and-parse-dumps/

Orhan, G. Y. (2020) *4 Reasons Why You Should Use Google Colab for Your Next Project.* Towardsdatascience. Retrieved March 8 From: https://towardsdatascience.com/4-reasons-why-you-should-use-google-colab-for-your-next-project-b0c4aaad39ed

Peng, F. & McCallum, A., (2006). Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4), pp.963-979.

Phyton. (2022). *AST - abstract syntax trees*. ast - Abstract Syntax Trees - Python 3.10.2 documentation. Retrieved March 8, 2022, from https://docs.python.org/3/library/ast.html

Python. (2022). Introduction to flask. Introduction to Flask - Python for you and me documentation. Retrieved July 17, 2022, from https://pymbook.readthedocs.io/en/latest/flask.html#:~:text=Flask%20is%20a%20web%20framework,application%20or%20a%20commercial%20website.

Phyton. (2022) *random — Generate pseudo-random numbers.* Phyton Docs. Retrieved March 8 From: https://docs.python.org/3/library/random.html

PyPI. (2022). *Transformers*. Retrieved June 14, 2022, from https://pypi.org/project/transformers/

Rajman, M., & Besançon, R. (1998). Text mining-knowledge extraction from unstructured textual data. In *Advances in data science and classification* (pp. 473-480). Springer, Berlin, Heidelberg.

Rashi, D. (2019) *Top 10 Python Libraries for Data Science*. Towardsdatascience. Retrieved March 8 From: https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266.

Roberti, M. (2019, January 23). FASTAI with transformers (Bert, Roberta, xlnet, XLM, Distilbert). Medium. Retrieved June 14, 2022, from https://towardsdatascience.com/fastai-with-transformers-bert-roberta-xlnet-xlm-distilbert-4f41ee18ecb2

SAS Institute. (2017). *Introduction to SEMMA*. SAS help center. Retrieved March 10, 2022, from https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm

seaborn. (n.d.). An introduction to seaborn. An introduction to seaborn - seaborn 0.11.2 documentation. Retrieved July 17, 2022, from https://seaborn.pydata.org/introduction.html#:~:text=Seaborn%20is%20a%20library%20for,explore%20and%20understand%20your%20data.

scikit-learn. scikit. (2022). Retrieved July 17, 2022, from https://scikit-learn.org/stable/

scipy. (2022). *Scipy*. Retrieved July 17, 2022, from https://scipy.org/

Shah, R. (2021) *How to Use Progress Bars in Python?* Analytics Vidhya. Retrieved March 8 From: https://www.analyticsvidhya.com/blog/2021/05/how-to-use-progress-bars-in-python/

Si, Y., & Roberts, K. (2018). A frame-based NLP system for cancer-related information extraction. In AMIA annual symposium proceedings (Vol. 2018, p. 1524). American Medical Informatics Association.

Singh, T. (2021). Natural language processing with spacy in python. Real Python. Retrieved July 17, 2022, from https://realpython.com/natural-language-processing-spacy-python/

Song, F. & W. Bruce, C., (1999). A General Language Model for Information Retrieval. *Association for Computing Machinery*, pp.316–321. From: https://doi.org/10.1145/319950.320022

Song, J., Lee, J. K., Choi, J., & Kim, I. (2020). Deep learning-based extraction of predicate-argument structure (PAS) in building design rule sentences. Journal of Computational Design and Engineering, 7(5), 563-576.

Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S., & Wang, G. (2018). Data processing and text mining technologies on electronic medical records: a review. *Journal of healthcare engineering*, *2018*.

Taylor, K. (2019). The retail apocalypse is far from over as analysts predict 75,000 more store closures.

Thavavel, V., & Sivakumar, S. (2012). A generalized framework of privacy preservation in distributed data mining for unstructured data environment. *International Journal of Computer Science Issues (IJCSI)*, *9*(1), 434.

TutorialTeacher. (2022). Phyton OS Module. Retrieved March 8, 2022, from https://www.tutorialsteacher.com/python/os-module

Vallantin, L. (2020). *Why you should not trust only in accuracy to measure machine learning performance*. Medium. Retrieved March 8, 2022, from https://medium.com/@limavallantin/why-you-should-not-trust-only-in-accuracy-to-measure-machine-learning-performance-a72cf00b4516

Wallach, H. M. (2004). Conditional random fields: An introduction. *Technical Reports (CIS)*, 22.

Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1, pp. 29-40).

Wu, P. J., & Lin, K. C. (2018). Unstructured big data analytics for retrieving e-commerce logistics knowledge. *Telematics and Informatics*, *35*(1), 237-244.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2021, March). Theoretical understandings of product embedding for e-commerce machine learning. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (pp. 256-264).

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629.

Zhang, F., Fleyeh, H., Wang, X., & Lu, M. (2019). Construction site accident analysis using text mining and natural language processing techniques. *Automation in Construction*, *99*, 238-248.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., ... & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1529-1537).

Appendices

Disclaimer Ethics Form

Office Record	***Receipt***
Date Received: Received by whom:	Student name: Hermanto Student number: TP054802 Received by: Date:

ACADEMIC RESEARCH ETHICS DISCLAIMER

Declaration about ethical issues and implications of research project/assignment proposals to be included on project/assignment application forms

Project/Assignment Title:

Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques

The following declaration should be made in cases where research project/assignment applicants for a particular project/assignment and the supervisor(s)/lecturer(s) for that project/assignment conclude that it is not necessary to apply for ethical approval for the research project/assignment.

We confirm that the University's guidelines for ethical approval have been consulted and that all ethical issues and implications in relation to the above project/assignment have been considered. We confirm that ethical approval need not be sought.

Hermanto		03/02/2022
Name of Research Project/Assignment Applicant	e-signature	Date
Dr. Dewi Octaviani	Dr. Dewi	3/3/2022
Name of Research Project Supervisor/ Assignment Lecturer	e-signature	Date

Academic Research Ethics Disclaimer Form ver. 3.0 (Dec 2015) Page 1 of 1

Fast-Track Ethics Form

Office Record	Receipt – Fast-Track Ethical Approval
Date Received:	Student name: Hermanto Student number: TP054802 Received by:
Received by whom:	Date:

APU / APIIT FAST-TRACK ETHICAL APPROVAL FORM (STUDENTS)

Tick one box (level of study):
- ☐ POSTGRADUATE (PhD / MPhil / Masters)
- ☑ UNDERGRADUATE (Bachelors degree)
- ☐ FOUNDATION / DIPLOMA / Other categories

Tick one box (purpose of approval):
- ☑ Thesis / Dissertation / FYP project
- ☐ Module assignment
- ☐ Other: _____

Title of Programme on which enrolled Computer Science Specialism in Data Analytics

Tick one box: ☑ Full-Time Study or ☐ Part-Time Study

Title of project / assignment: Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques

Name of student researcher: Hermanto

Name of supervisor / lecturer: Dr Dewi Octaviani

Student Researchers- please note that certain professional organisations have ethical guidelines that you may need to consult when completing this form.

Supervisors/Module Lecturers - please seek guidance from the Chair of the APU Research Ethics Committee if you are uncertain about any ethical issue arising from this application.

		YES	NO	N/A
1	Will you describe the main procedures to participants in advance, so that they are informed about what to expect?			✔
2	Will you tell participants that their participation is voluntary?			✔
3	Will you obtain written consent for participation?			✔
4	If the research is observational, will you ask participants for their consent to being observed?			✔
5	Will you tell participants that they may withdraw from the research at any time and for any reason?			✔
6	With questionnaires and interviews will you give participants the option of omitting questions they do not want to answer?			✔
7	Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?			✔
8	Will you give participants the opportunity to be debriefed i.e. to find out more about the study and its results?			✔

If you have ticked **No** to any of Q1-8 you should complete the full Ethics Approval Form.

		YES	NO	N/A
9	Will your project/assignment deliberately mislead participants in any way?		✔	
10	Is there any realistic risk of any participants experiencing either physical or psychological distress or discomfort?		✔	
11	Is the nature of the research such that contentious or sensitive issues might be involved?		✔	

If you have ticked **Yes** to 9, 10 or 11 you should complete the full Ethics Approval Form. In relation to question 10 this should include details of what you will tell participants to do if they should experience any problems (e.g. who they can contact for help). You may also need to consider risk assessment issues.

			YES	NO	N/A
12	Does your project/assignment involve work with animals?			✔	
13	Do participants fall into any of the following special groups? **Note that you may also need to obtain satisfactory clearance from the relevant authorities**	Children (under 18 years of age)		✔	
		People with communication or learning difficulties			
		Patients			
		People in custody			
		People who could be regarded as vulnerable			
		People engaged in illegal activities (eg drug taking)			
14	Does the project/assignment involve external funding or external collaboration where the funding body or external collaborative partner requires the University to provide evidence that the project/assignment had been subject to ethical scrutiny?			✔	

If you have ticked **Yes** to 12, 13 or 14 you should complete the full Ethics Approval Form. There is an obligation on student and supervisor to bring to the attention of the APU Research Ethics Committee any issues with ethical implications not clearly covered by the above checklist.

STUDENT RESEARCHER

Provide in the boxes below (plus any other appended details) information required in support of your application, THEN SIGN THE FORM.

Please Tick Boxes

I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU Research Ethics Committee.	✔
Give a brief description of participants and procedure (methods, tests used etc) in up to 150 words. For the project, the methods used to gather data from participants are outsourced from sites that are publicly available on the internet. The collection of data gathered from Kaggle is reliable data that are able use as a base for the machine learning algorithm modeling.	
I also confirm that: i) All key documents e.g. consent form, information sheet, questionnaire/interview are appended to this application.	N/A
Or ii) Any key documents e.g. consent form, information sheet, questionnaire/interview schedules which need to be finalised following initial investigations will be submitted for approval by the project/assignment supervisor/module lecturer before they are used in primary data collection.	N/A

E-signature: Print Name: Hermanto Date: 03/02/2022

(Student Researcher)

Please note that any variation to that contained within this document that in any way affects ethical issues of the stated research requires the appending of new ethical details. New ethical consent may need to be sought.
The completed form (and any attachments) should be submitted for consideration by your Supervisor/Module Lecturer

SUPERVISOR/MODULE LECTURER
PLEASE CONFIRM THE FOLLOWING:

	Please Tick Box
I consider that this project/assignment has no significant ethical implications requiring a full ethics submission to the APU Research Ethics Committee	✔
i) I have checked and approved the key documents required for this proposal (e.g. consent form, information sheet, questionnaire, interview schedule) Or	N/A
ii) I have checked and approved draft documents required for this proposal which provide a basis for the preliminary investigations which will inform the main research study. I have informed the student researcher that finalised and additional documents (e.g. consent form, information sheet, questionnaire, interview schedule) must be submitted for approval by me before they are used for primary data collection.	N/A

SUPERVISOR AND SECOND ACADEMIC SIGNATORY

STATEMENT OF ETHICAL APPROVAL (please delete as appropriate)

1) **THIS PROJECT/ASSIGNMENT HAS BEEN CONSIDERED USING AGREED APU/SU PROCEDURES AND IS NOW APPROVED**

2) **THIS PROJECT/ASSIGNMENT HAS BEEN APPROVED IN PRINCIPLE AS INVOLVING NO SIGNIFICANT ETHICAL IMPLICATIONS, BUT FINAL APPROVAL FOR DATA COLLECTION IS SUBJECT TO THE SUBMISSION OF KEY DOCUMENTS FOR APPROVAL BY SUPERVISOR (see Appendix A)**

E-signature *Dr. Dewi* Print Name Dr. Dewi Octaviani Date 3/3/2022
(Supervisor/Lecturer)

E-signature… … … … … … … … … … … … Print Name… … … … … … … … … … … … Date… … … …
(Second Academic Signatory)

Office Record	Receipt – Appendix A (Fast-Track Ethics Form)
Date Received:	Student name: Hermanto Student number: TP054802 Received by:
Received by whom:	Date:

APPENDIX A
AUTHORISATION FOR USE OF KEY DOCUMENTS

Completion of Appendix A is required when for good reasons key documents are not available when a fast track application is approved by the supervisor/module lecturer and second academic signatory.

I have now checked and approved all the key documents associated with this proposal e.g. consent form, information sheet, questionnaire, interview schedule

Title of project/assignment

Name of student researcher Hermanto

Student ID: Intake:

E-signature… … … … … … … … … … … … Print Name… … … … … … … … … … … … Date… … …….. …
(Supervisor/Lecturer)

PPF (Photostat Copy)

Title: Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques

1. Introduction

Throughout our daily life, people ought to buy something whether it is grocery or daily necessity to fulfill daily needs. Thus surrounding the environment there will always be at least a grocery store that sells daily necessities. But as time progresses people find it a hassle to buy items at the grocery store or market. It is a hassle to drive their way to the store even though it's nearby, and you will always need to consider time, weather conditions, parking car, lining-up, and so on. There are too many disadvantages which are why there exists online retail marketing. Online retail service is one of the most popular topics as they are rapidly expanding due to people being prevented from going out of their house due to COVID-19. According to OECD the sales of online retail by 30% in April 2020 compared to April 2019. The online retail service allows them to have their desired items to be delivered right in front of their house, at the desired time and date without much hassle at all. As a result, many people are willing to use online retail services.

Recently, from the perspective of the retail marketing service in Indonesia, the marketer encountered a problem. The addresses entered by the customer are often not clear or incomplete thus the system cannot recognize the address when it passes for geocoding. Therefore, the aim of the project is to apply Data Mining Technique to help the extraction of the address entered by customers, predicting the point of address and street. and completing the incomplete words.

2. Problem Statement

The address element extraction purpose is to ensure the address elements that were passed for geocoding can be geocoded. According to Chakravarty (2018), geocoding allows address standardization which makes it easier for finding the exact address and coordination that can be pointed on the map. Thus, using geocoding can be useful to perform analysis of a region or market that can be used for drawing insightful information and deciding.

The process of geocoding simply needed the standardized version of address however that is not easily achieved as most of the input entered by customers are either incomplete or unstructured thus there was a problem existing between passing the address to standardization of the address before getting the geocoding process. According to Buzaianu (2020), it was common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. Therefore, a data analyst is assigned to solve this problem, correcting and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding.

3. Aim

The aims of the research to implement the data mining technique extracting address elements from the unstructured text

4. Objective

1. To determine sui techniques for extracting unstructured e-commerce data.

2. To implement data mining techniques for unstructured addresses for extracting a point of interest and street address

3. To implement data mining techniques to develop a spelling correction model and standardization of the address to complete and allow geocoding analysis.

4. To evaluate the spelling correction model and the geocoding analysis result.

5. Literature Review

RNN Model

The study of Li and his team (2020) predicted the English version from the Chinese address. It uses a recurrent neural network (RNN) for address segmentation. The method of segmentation is first by inputting the Chinese address sequence which will then be vectorized through the neural network. After that, the Viterbi algorithm is used for the last segmentation and tagging. It is one of the most powerful algorithms as it's the only algorithm that has an internal memory making it powerful and robust (Donges, 2019). The way RNN works is similar to the feed-forward neural network and sequential data.

BERT Model

Another research of information extraction made by Gupta and Nishu (2020), their method of extraction is language-based, done by the Bidirectional Encoder Representations from Transformers (BERT) model where it assigns tokens to each word which later be used for the Named Entity Recognition process. The BERT model is essentially a model published by researchers of Google AI language (Horev, 2018). The model makes use of the "transformer" which learns the relation between words in a text.

CRF Model

Information extraction is done by Zhang and his team (2008), extracting keywords automatically using the Conditional Random Fields (CRF) model, using two approaches. One of the approaches is keyword extraction where the words are analyzed to identify the words that are relational with each other based on their frequency rate and word lengths. The second approaches are keyword assignments where the keywords are specifically chosen from their vocabulary terms and are classified according to the content which will be related back to the vocabulary terms of the keywords.

N-Gram Model

Song and Croft (2015) define N-Gram Model as a general statistical language that are usually used for retrieving information. Most of the time the statistical language are used to predict the key elements and sequences of words by relying on a large corpus of documents. According to researches (Eugene, 1994 Song and Croft, 2015 Ahmed, William De Luca and Nürnberger, 2009), The statistical language model has been used for recognizing the voice of different people and parsing sentences synthetically. Their research also did comparisons between other language models and it was statistically proved that the statistical model the team developed is better than Ponte and Croft language model.

The proposed model will be divided two main parts. The first part would be Named Entity Recognition Model to recognize the tokenized words and to correctly predict the specific elements. Following Figure 5, the visualization of Name Entity Recognition can be seen as it utilizes the BERT model to predict the range of tokens that are the answer, meaning it was framed as the QA Problem Model.

The second part of the model would be developing a spelling correction model which is necessary to increase the accuracy of prediction as the data often has incomplete words in both elements. Thus the model is deployed to correct the labeled token which is not complete. The data will be labeled as shown in Table 4. All of the data that is labeled as 0 will be analyzed with the help of a 2-gram and 3-gram model to know the pattern of subsequent words, increasing the accuracy in prediction.

The feature of the model is the output of the prediction of the model is a human-readable address that has no incomplete words in it as well as having the element of the point of interest and the street address as well. It is a feature having human-readable addresses as output because the main requirement when it comes to addresses that are to be passed for address Geocoding with API is the human-readable address itself. This solves the problem of getting inaccurate coordination or error from the process geocoding due to the address not existing or incomplete address being provided.

PSF (Photostat Copy)

Title: Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques

1. Introduction

Throughout our daily life, people ought to buy something whether it is grocery or daily necessity to fulfill daily needs. Thus, surrounding the environment there will always be at least a grocery store that sells daily necessities. But as time progresses people find it a hassle to buy items at the grocery store or market. It is a hassle to drive their way to the store even though it is nearby, and you will always need to consider time, weather conditions, parking car, lining-up, and so on. There are too many disadvantages which are why there exists online retail marketing. Online retail service is one of the most popular topics as they are rapidly expanding due to people being prevented from going out of their house due to COVID-19. According to OECD the sales of online retail by 30% in April 2020 compared to April 2019. The online retail service allows them to have their desired items to be delivered right in front of their house, at the desired time and date without much hassle at all. As a result, many people are willing to use online retail services.

Recently, from the perspective of the retail marketing service in Indonesia, the marketer encountered a problem. The addresses entered by the customer are often not clear or incomplete thus the system cannot recognize the address when it passes for geocoding. Therefore, the aim of the project is to apply Data Mining Technique to help the extraction of the address entered by customers, predicting the point of address and street. and completing the incomplete words.

2. Problem Context

The address element extraction purpose is to ensure the address elements that were passed for geocoding can be geocoded. According to Chakravarty (2018), geocoding allows address standardization which makes it easier for finding the exact address and coordination that can be pointed on the map. Thus, using geocoding can be useful to perform analysis of a region or market that can be used for drawing insightful information and deciding.

The process of geocoding simply needed the standardized version of address however that is not easily achieved as most of the input entered by customers are either incomplete or unstructured thus there is a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it is common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear

address, making the address cannot directly be passed for geocoding. Therefore, a data analyst is assigned to solve this problem, correcting and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding.

3. Rationale

Customers tend to enter incomplete or unstructured, thus there is a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it is common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. Therefore a data analyst is assigned to solve this problem, correcting and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding

4. Potential benefits

This research will benefit greatly for the online retail sectors and marketing, accurately predicting the element of incomplete address that will play an important role in the implementation of geocoding to obtain the geographic coordinates to deliver the parcel, ensuring efficient and high-quality customer satisfaction with the quick arrival of the parcel. The common occurrence for customers mistyping or entering incomplete addresses encourages the demand for the proposed system and for the online retail company that is planning for the implementation of geocoding. The inventory department that oversees finding parcel addresses will utilize the system, saving human resources and the need to manually find the accurate address of parcels.

4.1 Tangible benefits

i. Lower the costs of labour and and time with the improvement of efficiency as machine will do the task automatically
ii. Allow the company to do analysis on the address data as the address is already standardized into required formats
iii. Allow the company to identify incorrect address and make decision from there

4.2 Intangible benefits

i. It increases customer experience as there will not be any incorrect address being sent to the system, thus parcel or item will arrive at the correct addresses.
ii. It gives a much more accurate results compared to being done by employee and able to reduce the labour
iii. Lower the risk of sending parcel to the incorrect addresses

5. Target users

E-Commerce will require customers to enter or type down their addresses information when they are buying things from their website. All the addresses that was sent by customers will be reviewed by someone to check the location and standardized according to the required formats manually which is inefficient and a lot of work. The model developed will automatically do all the things mentioned automatically and have stable rate of accuracy.

6. Scope and objectives

6.1 Aim

Customers tend to enter incomplete or unstructured, thus there is a problem existing between passing the address to standardization of the address before getting into the geocoding process. According to Buzaianu (2020), it is common for customers to make mistakes such as misspelling, giving an incomplete address, and unclear address, making the address cannot directly be passed for geocoding. This is why a data analyst is assigned to solve this problem, correcting and completing the address entered by the customer and extracting the necessary elements to be passed for geocoding

6.2 Objectives

i. To determine suitable techniques for extracting unstructured e-commerce data.

ii. To implement data mining techniques for unstructured addresses for extracting a point of interest and street address.

iii. To implement data mining techniques to develop a spelling correction model and standardization of the address to complete and allow geocoding analysis.

iv. To evaluate the spelling correction model and the geocoding analysis result.

7. Deliverables - Functionality of the proposed system

The requirements for the machine learning algorithm is to be able to Extract two elements from the raw address which are POI and street address have the accuracy of 70% or above for correcting the spelling and does not contain abbreviation

8. Nature of Challenges

The challenge would be developing two main models for the machine learning algorithm. The first challenge would be developing the Named Entity Recognition Model to recognize the tokenized words and to correctly predict the specific elements. Following Figure 5, the visualization of Name Entity Recognition can be seen as it utilizes the BERT model to predict the range of tokens that are the answer, meaning it was framed as the QA Problem Model.

The second part of the model would be developing a spelling correction model which is necessary to increase the accuracy of prediction as the data often has incomplete words in both elements. Thus, the model is deployed to correct the labeled token which is not complete.

9. Overview of the Proposal

The Chapter 1 will provide the overall description of the report, including the objectives and aim of the project, the benefits, and the challenges. In the chapter 2, deep literature review is written to conduct research and gain knowledge of the project. The knowledge domain research includes,

Text mining of unstructured data, E-commerce, Prediction model, and machine learning for e-commerce. For chapter 3, it discusses the technical part of the project which include the computer language that will be used to build the machine learning model, the IDE and the basic requirement to run the IDE or build the model. For chapter 4, methodology is discussed in detail, first there will be comparison between two model which is CRISP-DM and SEMMA, then decision will be made between the two which one is chosen for the project. Afterward, in detail describe the methodology of CRISP-DM and how will it be applied in the project. Examples are also provided in table and figures to show how it was done or to give an overall expectation. Lastly, the chapter 5 will sum up the whole report and contain reflections of the author.

10. Programming language chosen

R and Python are open-source programming languages with a massive community. New libraries are added continuously to their respective catalog. R is used for statistical assessment on the identical time as Python gives a more favored approach to statistics science. R and Python are great in terms of programming language oriented in the direction of statistics science. Learning both languages are the perfect solution, but it is time consuming. Python is a favored-motive language with a readable syntax. R, however, is built thru manner of method of statisticians and encompasses their language.

10.1 R Language

Scholars and statisticians had been growing R for over 20 years. R is now one of the richest languages for information analytics. The Open-Source Repository (CRAN) has approximately 12,000 packages. You can locate libraries for any evaluation you need to perform. The library makes R the first-class desire for statistical evaluation, specifically for specialized analytical tasks. The maximum superior distinction among the R and different statistical merchandise is the output. R has superb reporting tools. RStudio comes with the Knitr library. Xie Yihui wrote this package. He made the file trivial and elegant. Easily speak effects in displays or documents.

10.2 Python

Python can do lots of the equal component as R: information processing, development, feature-selective net scraping, applications, and extra. Python is a device for deploying and enforcing device mastering at scale. Python code is less difficult to keep and extra dependable than R. few years ago; Python didn't have many libraries for information evaluation and device mastering. Recently, Python is catching up and gives cutting-edge API for device mastering or Artificial Intelligence. Most of the information technology activity may be executed with 5 Python libraries: Numpy, Pandas, Scipy, Scikitlearn and Seaborn. Python, on the alternative hand, makes replicability and accessibility less difficult than R. In fact, in case you want to apply the effects of your evaluation in a software or website, Python is the first-class desire.

10.3 Summary

For the project since I am planning to do data science related project, as well as building machine learning algorithm relying on libraries, I will be using phyton as it is more suited language to be used.

11. IDE (Interactive Development Environment) chosen.

Google Colab is a free IDE available on the internet. It is a product of Google studies and is primarily based totally on Jupyter. Colab is a brilliant device for novice and expert users, nearly all essential libraries are preinstalled with it so that you there is no need to deploy library one by one. Colab's notebook documents are saved for your google drive, so that they may be accessed from everywhere you want. It additionally permits you to percentage your pocketbook together along with your colleague without even downloading it, that is significantly the high-quality function for many. Apart from this it additionally offers loose GPU and TPU in your paintings and that makes it perfect for Deep Learning and Machine Learning projects. Main advantage of Google Colab is the convenience of it and not require tons of resources to run the code as all of the resource are provided by google cloud service. Other than that, Colab comes with the libraries thus it will save time from installing libraries.

12. Libraries chosen / Tools chosen

12.1 pandas

Pandas is the most popular open-source Python package for data processing/data analysis and machine learning tasks. It is built on top of another package called Numpy which supports multidimensional arrays. As one of the most popular data processing packages, Pandas works well with many other data processing modules in the Python ecosystem and is included in every Python distribution, from those that usually ship with operating systems to commercial vendor distributions like ActiveState ActivePython.

12.2 string

Python String module contains some constants, utility function, and classes for string manipulation. Since the project involved textual data, the string library will be used to manipulate the textual data, such as removing whitespace, punctuation, or separating the string.

12.3 tqdm

Tqdm is a library in Python that is used for developing Progress Meters or Progress Bars. tqdm were given its call from the Arabic call taqaddum which means `progress`. Implementing tqdm may be completed results easily in our loops, capabilities or maybe Pandas. Progress bars are quite beneficial in Python as it permits to look if the Kernel continues to be running and Progress Bars are visually attractive to the eyes. Other than that, it offers Code Execution Time and Estimated Time for the code to finish which might assist at the same time as running on large datasets

12.4 random

The random module of the phyton library allows them to generate random numbers. The random number is generated according to the generator seed. This module can be used to perform random operations such as generating random numbers, printing random values for lists or strings, etc.

12.5 Json

The JSON module is specifically used to transform the python dictionary above right into a JSON string that may be written right into a file. While the JSON module will convert strings to Python

datatypes, typically the JSON capabilities are used to examine and write without delay from JSON files.

12.6 os

The OS module in Python affords features for growing and casting off a directory (folder), fetching its contents, converting, and figuring out the modern-day directory, etc. This module **offers** a **transportable manner** of **the use of running device structured** functionality.

12.7 ast

The ast module helps Python applications handle Python abstract syntax grammar trees. The abstract syntax itself may change from release to release of Python. This module helps you to know programmatically what the current grammar looks like.

13. Operating System chosen

Windows 10 is the operating system chosen for the project as it supports most of the application available. Especially since I am using Windows 10 as default operating system, and Google Colab as IDE, Windows 10 is great choice as Google Colab is web-based and does not have strict requirements.

14. Hardware Requirement

Following are minimum requirement to run Google Colab:

Name	Requirement	Current
Processor	**Intel Core i5-4590**	**AMD Ryzen 9 4900H**
Graphics Card	**NVIDIA GeForce GTX 970**	**NVIDIA GeForce RTX 2060**
Memory	**4 GB**	**16 GB**
Available Space	**2 GB**	**100+ GB**

15. Software Requirement

Following are minimum requirement to run Google Colab:

Name	Requirement	Current
Operating System	**Windows 7**	**Windows 10**
Software	**Any Internet Browser**	**Google Chrome**

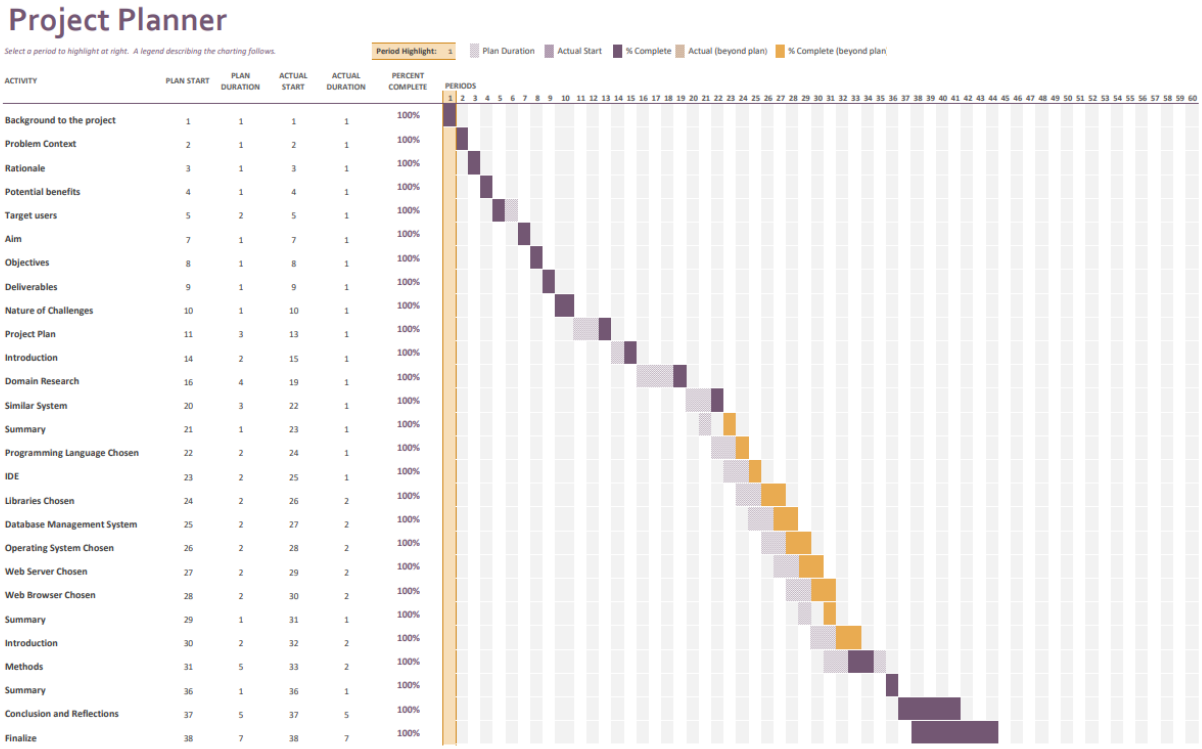
16. Web Server chosen

Django is a Python net framework made for experts users as it encourages fast improvement and clean, pragmatic design. Built through skilled developers, it looks after lots of the problem of net improvement, so that you can recognition on writing your app while not having to reinvent the wheel. It`s loose and open source.

17. Web browser chosen

Chrome is designed to be the quickest internet browser. With one click, it masses internet pages, a couple of tabs, and packages with lightning speed. Chrome is equipped with V8, a quicker and extra effective JavaScript engine. Chrome additionally masses internet pages quicker using the WebKit open supply rendering engine.

Gantt Chart


Project Planner
Select a period to highlight at right. A legend describing the charting follows.
Period Highlight: 1
Plan Duration
Actual Start
% Complete
Actual (beyond plan)
% Complete (beyond plan
ACTIVITY
PLAN START
PLAN DURATION
ACTUAL START
ACTUAL DURATION
PERCENT COMPLETE
PERIODS
Background to the project 1 1 1 1 100%
Problem Context 2 1 2 1 100%
Rationale 3 1 3 1 100%
Potential benefits 4 1 4 1 100%
Target users 5 2 5 1 100%
Aim 7 1 7 1 100%
Objectives 8 1 8 1 100%
Deliverables 9 1 9 1 100%
Nature of Challenges 10 1 10 1 100%
Project Plan 11 3 13 1 100%
Introduction 14 2 15 1 100%
Domain Research 16 4 19 1 100%
Similar System 20 3 22 1 100%
Summary 21 1 23 1 100%
Programming Language Chosen 22 2 24 1 100%
IDE 23 2 25 1 100%
Libraries Chosen 24 2 26 2 100%
Database Management System 25 2 27 2 100%
Operating System Chosen 26 2 28 2 100%
Web Server Chosen 27 2 29 2 100%
Web Browser Chosen 28 2 30 2 100%
Summary 29 1 31 1 100%
Introduction 30 2 32 2 100%
Methods 31 5 33 2 100%
Summary 36 1 36 1 100%
Conclusion and Reflections 37 5 37 5 100%
Finalize 38 7 38 7 100%

Project Log Sheet

DATE OF MEETING	TIME FROM	TIME-TO	SEMESTER	ITEM FOR DISCUSSION	RECORD OF DISCUSSION	ACTION LIST	STATUS
30/12/2021	02:30 PM	03:00 PM	1	My PPF contains ambiguity and we discuss the things that are unclear as well. Some changes were done to improve the documentation and also showed the dataset that I have as well as explaining the proposal of the project, what I am going to do.	The title was changed from Online Retail Marketing: Address Elements Extraction from unstructured text using Data Mining Technique to Extraction of Unstructured Textual Data in E-Commerce using Data Mining Techniques. Objectives are to have specific areas instead of broad areas.	Proceed with PSF and compare and contrast CRISP-DM and SEMMA and choose one methodology between the two methods. This is completed by the next meeting so there is something to be handed	Your supervisor has reviewed it
09/02/2022	10:00 PM	11:00 pm	1	Investigation Report advice including what should we include in each chapter in the report and what should we avoid doing when writing the report and we will have to submit draft of IR for the next meeting and wait for comment from supervisor and based on the comment, we will make changes on the Investigation Report The meeting discussion include the discussion of investigation report where the supervisor provide advice. The advice consist of do and do not, things we should avoid when writing our report	We should include more content in the literature review depending on the title that was chosen, each of them should contain expertise knowledge that are related to the project. For methodology part we should also explain things and steps in details and avoid general discussion.	For the next meeting, we are told to prepare investigation report draft that contain all of the chapters. The draft changes is prepared based on the comments given by the supervisor.	Your supervisor has reviewed it
13/02/2022	09:00 pm	09:30 pm	1	Previously, supervisor tasked us to submitt investigation report draft. The report that was provided, comments were provided by supervisor based on the draft. The comments contain the mistakes we have made and chapters that we have not added in to the draft.	Missing Acknolwedgement, Table of Contents, List of Tables, For the Chapter 1 missing overveiw of the report and project plan, For Chapter 2 missing , For Chapter 3 need to add more details on the programming, tools, and IDE, For Chapter 4 incomplete methodlogies, lack comparison and need to provide more detail for the data preparation etc. Lastly, need to add Chapter 5 and include references and appendices	The mistakes mentioned should be corrected or be filled. For the next meeting we should provide finalized investigation draft which include all chapters and is properly sorted	Your supervisor has reviewed it
03/03/2022	10:00 PM	10:30 PM	1	The items being discussed is Investigation report for the finalized draft. The comments regarding all of the chapters of the investigation report were given by supervisor in detail.	Correct the mistakes on the cover page, project plan. For the similar model's section, the authors should be in order based on the year, as for the Literature Review lack contents thus need to add more, sort out the chapter properly, and as well as correcting spelling error.	The investigation report should be finalized based on the comments given by the supervisor and we should be able to submit the investigation report on time based on the deadline provided in moodle system	Your supervisor has reviewed it