National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cybersecurity

Yejin Lee
Student ID: X20227809

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Yejin Lee
Student ID:	X20227809
Programme:	MSc in Cyber Security
Module:	Academic Internship
Supervisor:	Michael Pantridge
Submission Due Date:	14/08/2023
Project Title:	Preventing Remote control Smishing on Android OS
Word Count:	513 **Page Count:** 5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Yejin Lee
Date:	30 July 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Yejin Lee
Student ID: X20227809

1. Introduction

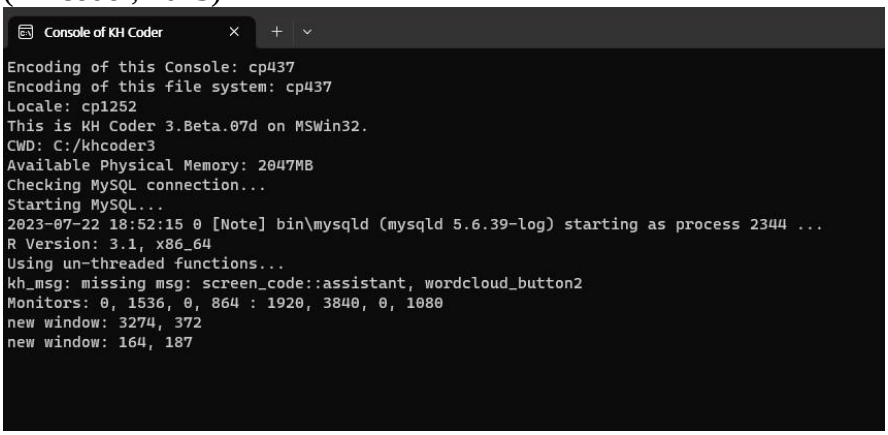
The configuration manual describes the system requirements for research and the setup environment for text mining and machine learning testing.

2. Systems requirement

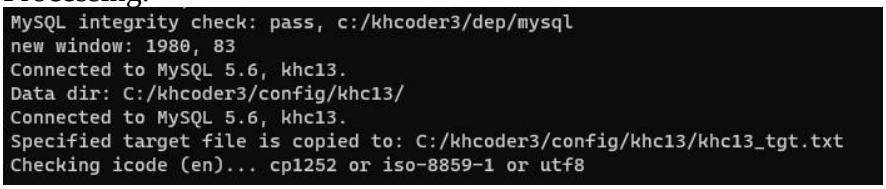
- Text mining tests require KH Coder (v3.0)
- Case analysis mobile device, Android 8 , 18 (Samsung)
- Jupyter Notebook (v6.5.4)
- Python (v 3.11.4)
- Scikit learn Library
- Anaconda (v2.3.2)

3. Text mining

Convert a dataset (CSV) file to a TXT file. After, unzip the file name ‘khcoder-3b07d.exe’. (KHcoder, 2023)



Run KHcoder3 exe file. After creating a new project, upload the txt file. And, execute Pre-Processing.



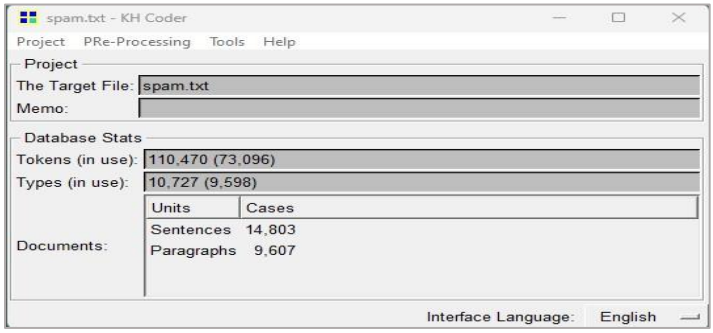


Figure 1 Upload public dataset txt file to KH coder

4. Test Code

The code for Naive Bayes machine learning was written in Python. Also, when testing, we can use the sklearn ConuntVectorizer library to classify messages. (scikitlearn, 2023)

```
# import pakages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#import Library
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
```

Then, the public data set and the data set collected in the research are merged, produced in csv format, and used in the Python code. (VanderPlas, J, 2016)

```
# Dataset Upload
df = pd.read_csv("dataset.csv")


#Show the all inspect data
df


# Categorizing
df.groupby('Label').describe()



#turn spam/ham into numerical data, creating a new column called 'spam'
df['spam'] = df['Label'].apply(lambda x: 1 if x== 'spam' else 0)
```

In this research, we split the training and test datasets for each 0.7, 0.3.

```
#turn spam/ham into numerical data, creating a new column called 'spam'
df['spam'] = df['Label'].apply(lambda x: 1 if x== 'spam' else 0)


#create train and test split
x_train, x_test, y_train, y_test = train_test_split(df.Text, df.spam, test_size =0.3)
x_train.describe()

#find word count and store data as a matrix
cv = CountVectorizer()
x_train_count = cv.fit_transform(x_train.values)
x_train_count
x_train_count.toarray()

#train model
model = MultinomialNB()
model.fit(x_train_count, y_train)
```

Afterwards, we can test by directly inserting a message into the test, and check the test model score.

```
#We can pre-test non-spam messages individually
email_ham = ["Please enter a example non-spam message"]
email_ham_count = cv.transform(email_ham)
model.predict(email_ham_count)

#We can pre-test spam messages individually
email_spam =["Please enter a example spam-message"]
email_spam_count = cv.transform(email_spam)
model.predict(email_spam_count)

#test model
x_test_count = cv.transform(x_test)


#test model
model.score(x_test_count, y_test)
```

Also in this study, we added the following code to prevent LaPlace smoothing.

```
#Create a custom CountVectorizer class with Laplace smoothing
class LaplaceCountVectorizer(CountVectorizer):
    def __init__(self, alpha=1.0, **kwargs):
        super().__init__(**kwargs)
        self.alpha = alpha

    def fit(self, raw_documents, y=None):
        super().fit(raw_documents, y)
        self.feature_log_prob_ = np.log((self.alpha + self.transform(raw_documents).sum(axis=0))
        / (self.alpha * self.transform(raw_documents).sum()))

    def transform(self, raw_documents):
        X = super().transform(raw_documents)
        X.data += self.alpha
        return X


cv = LaplaceCountVectorizer()
x_train_count = cv.fit_transform(x_train.values)



class LaplaceMultinomialNB(MultinomialNB):
    def __init__(self, alpha=1.0, **kwargs):
        super().__init__(alpha=alpha, **kwargs)


model = LaplaceMultinomialNB(alpha=1.0)
model.fit(x_train_count, y_train)
```

References

“Software for text mining” KH Coder Version 3.0, 2022. [Online]. Available: https://khcoder.net/en/. [Accessed: 22-July-2023].

“Count Vectorizer Library” Sklearn feature extraction 3.0, 2022. [Online]. Available: https://scikit-learn.org/ . [Accessed: 22-July-2023].

VanderPlas, J., 2016. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.".