ABSTRACT

Spam Checker Analysis systems are basically binary class or multi-class classification system that classifies feedback of customers into various Specific classes. It has become an industry on its own. There are dozens of notable internet companies, which can be referred as app companies who are doing customer feedback analysis for other, often much larger companies. Many feedback companies like Freshdesk and Nebula do analysis to various internet age companies like Amazon , Google ,Microsoft . Multi-class classification of texts is the challenging task in the field of Machine Learning. Our work focus at the task of six class classification of feedbacks received from different customers. Our corpus is categorized into six class classification namely comment, request, bug, complaint, meaningless and undetermined. The training and testing sets are generated using 10-fold Cross-validation method. We have achieved an accuracy of 64.70% using Random Forest algorithm. However, he baselines accuracy achieved by us is 53.42% using Gaussian Naïve Bayes algorithm.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
API	APPLICATION PROGRAM INTERFACE
GUI	GRAPHICAL USER INTERFACE
NB	NAÏVE BAYES
NLP	NATURAL LANGUAGE PROCESSING
SVM	SUPPORT VECTOR MACHINE

CHAPTER 1

SPAM CHECKER ANALYSIS

Introduction:

Spam Checker is a marketing term that describes the process of obtaining a spam opinion about a business, product or service. Spam checker is so important because it provides marketers and business owners with insight that they can use to improve their business, products and/or overall spam experience. Spam analysis measures how happy spam are with a company's products and services. Feedback analysis provides companies with feedback about everything from products to the buying process to support. Most organizations combine this powerful data with other forms of spam to create actionable intelligence about the entire spam journey. Spam is the one thing that gives a business a clearer view of how it is doing. Proper analysis provides a business with a better view of what it has to change, what it has to improve on, and what it has to do, to retain and grow revenue and profit.

Each customer review counts. But it is practically impossible to go through each customer manually. Hence a robust and an efficient system is necessary for the classification of the spam checker of a commodity. We aim to classify a feedback into six classes, namely comment, request, bug, complaint, meaningless and undetermined (further discussion has been done in chapter 4). Using past data we train our system to learn and hence use that acquired knowledge to assign a new spam checker into its suitable class. The whole process makes the evaluation of a product in the future market easy

This thesis consists of 6 chapters. chapter 1 introduces us to the problem statement while chapter 2 elaborates the related work done in this field. chapter 3 discusses the tools that we have used in the project. chapter 4 deals with the implementation of our work in detail. chapter 5 displays the observations obtained and consequently.

1.1. Motivation

One of the main reasons we decided to work together was our passion for natural language processing (NLP) . While the domain of NLP has various sub-domains, we were interested in putting our efforts on the multi-class classification field. So we embarked on a journey to achieve the highest possible accuracy in a spam checker analysis system. The theme of the project was inspired by the fact that it is not possible for companies to go through each of the spam checker for their products manually and decide what should be their priority. Instead, we designed a system which will classify the customer reviews to appropriate classes efficiently.

1.2. Problem statement

As mentioned in the introduction, the objective of the project is to analyze and classify a spam checker into suitable classes. Basic idea is to build an intelligent multi-class classification system.

The most challenging parts in Spam Analysis are:

- It is important that the feedback tool provides us with analysis and automated flows, or else, we could be stuck patch-working different solutions to run one simple spam checker process or worse, do all that manually.

- Another spam checker challenge comes in the final, and arguably, most important step of any checker process. And one can guess, any manual processes will delay the steps in spam checker cycles. It is important to act upon the results of your checker analysis. This means acting upon the good and the bad.

- Most often, because the process requires manual work, companies tend to pay less attention to the good ham and just focus on acting to remedy the bad spam. While this may be enough to get by, it is not the best practice.

CHAPTER 2

RELATED WORK

There has not been a lot of work done in the field of spam checker analysis. However, a lot of research work has been done on multi-class classification.

One other work done is precision of multi-class classification methods for Support Vector Machines by Hongjian et. al [2]. They have used multi-support vector machines together.

They have experimented using one-against-one SVM, one-against-the-rest SVM and binary tree SVM. They have drawn that precision of Binary Tree SVM is better than that of one-against the-rest SVM, and in all types of Binary Tree SVM, precision of Best Balanced Binary Tree SVM is best, precision of Worst Balanced Binary Tree SVM is worst.

Pingpeng et. al has presented their work MSVM-kNN: Combining SVM and kNN for Multiclass Text Classification [3]. According to their paper, SVM can not identify categories of documents correctly when the texts are in cross zones of multi-categories, kNN cannot effectively solve the problem of overlapped categories borders. However, their experiments have shown that using SVM for identifying category borders, and then using kNN for classifying documents among the borders improve performance. Their exeperimental results show that their approach Multi-class SVM-kNN (MSVM-kNN) performs better than SVM or kNN.

Gamon, M. [4] describes in his paper on sentiment classification on spam checker data that large feature vectors in combination with feature reduction can train linear support vector machine that can achieve high classification accuracy. The paper suggests that the addition of deep linguistic analysis features to a set of surface level word n-gram features contributes consistently to classification accuracy in this domain. An another research by Pal, M. on Random forest classifier for remote sensing classification [5] shows that random forest classifier performs equally well to
SVMs in terms of classification accuracy and training time. Another research, an improved random forest classifier for multi-class classification by Chaudhary et. al [7] showed a greater increase in disease classification accuracy as 97.80% as compared to Neural Networks, Logistic Regression and SVM as 92.20%, 94.80%, 95.70% respectively.

2.1 IDE & Text Editors

An Integrated Development Environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and debugger.

2.1.1 Python Note book

Python is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language , that offers introspection, rich media, shell syntax, tab completion, and history. I Python provides the following features:

- Interactive shells (terminal and Qt-based).
- A browser-based notebook with support for code, text, mathematical expressions, inline plots and other media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into one's own projects.

2.2 Languages & Technologies

2.2.1 Python:

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.

2.3 Python Lybraries

2.3.1 SK Learn:

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

2.3.2 Pandas:

Pandas is an extension to the Python programming language, adding support for data Manipulation and analysis. In particular it offers data structures and operations for manipulating numerical tables and time series.

2.3.3 Numpy:

NumPy is an extension to the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created.

2.3.4 Gensim:

Genism is a robust open-source vector space modeling and topic modeling toolkit implemented in Python.It uses NumPy, SciPy and optionally Python for performance. Genism is specifically designed to handle large text collections, using data streaming and efficient incremental algorithms, which differentiates it from most other scientific software packages that only target batch and in memory processing.

2.3.5 Pickle:

Python pickle module is used for serializing and de-serializing a Python object structure.

CHAPTER 3

IMPLEMENTATION

3.1 Introduction

In Implementation part we have combined all the things we had done in earlier phases of our project development process. We have used all the technologies described in requirement analysis part to achieve all the functionalities of our system. We used supervised learning approach of machine training approach to train our system to achieve better classification.

Our approach of implementing Spam Analysis System was divided into 4 parts, namely:

- Corpus Collection
- Feature Extraction
- Splitting of Dataset
- Training and Modeling

3.1.1. Corpus Collection

In a joint ADAPT (ADAPT Venture)-Microsoft research project, representative real world samples of customers' feed-backs has been released jointly by Microsoft and IJCNLP (International Joint Conference on Natural Language Processing).

About the corpus

Syntax:

dictionary = gensim.corpora.Dictionary(mails)

corpora means collection of data i.e,mails

Number of documents = 3565

Number of tokens = 45135

Number of classes = 6

Class	No. of documents	No. of tokens
Bug	88	1553