COMP 6721: Artificial Intelligence, Project 2 Report

Aria Adibi, 40139168

Abstract

In this project the title of the posts of a popular site, *HackersNews*, of the year 2019 is to be predicatively classifies into post types, given the data of the year 2018. The aim is to familiarize ourselves with the following:

1. How to parse texts and extract needed information in the context of Natural Language Processing
2. To implement Naive Bayes algorithm, one of the easier (generative) classifiers in supervised learning algorithms.
3. To investigate some of the easy modifications for performance improvements.

Contents

List of Tables

1 Introductions and technical details

1.1 Introduction

Following information of posts in a popular site, HackersNews, for the years 2018 and 2019 is given.

Object ID	Title	Post Type	Author	Created At	URL	...
...	Points	Number of Comments				

The chosen supervised model is Naive Bayes, which has to predict the Post Type given the Title. The training data is the ones created at 2018 and the test data is the 2019 ones.

My code first parse the given `.csv` file and extract the useful information, with the help of `NLTK` package.

Then the Naive Bayes algorithm is implemented. At the end some tweaks are investigated to study their effects on performance and accuracy.

1.2 Technical details

In the provided `.zip` file you will find the following files:

- `README.txt`
  This `.txt` file provides the instructions needed to run the program.
- `main_v1.py`
  The main (runnable) python module which is to be executed.
- `Technical Report`
  This file, which reports and give analysis of the project.
- `Expectation of Originality`
  A signed form for the purpose of originality of the work.

2 Explanation of the parsing and extraction part

2.1 Observation

By manually observing the data, one can see that the data has no missing values, therefore no `imputation` is needed. Also the data seems balanced with no mistake.

2.2 The methodology

After reading the `.csv` file and separating the columns, the additional information is removed and data is parted in two sets: Training and Testing. With the help of the `NLTK` package then the data in Title column was `tokenized`. One could tokenize proper names such as "Computer Science" as one item with the tags associated in the when NLTK parses the text. At the same time "stop words" are removed and the frequency of words and their post-type-conditional frequency is accumulated. At the end a smoothed probability (for resolving zero probability issue) is associated.

The smoothing function used is *Additive Smoothing*, also called *Laplace smoothing*, with its "psedocount" set to 0.5.

3 Results and their analyzation

3.1 Naive Bayes Model

Bayes Theorem: *Let* $\{B_1,\ B_2,\ \ldots, B_n\}$ *be a partition of the sample space* S *of the experiment. If for* $i = \ 1,\ 2,\ \ldots,\ n,\ \ P(B_i) > 0$*, then for any event* A *of* S *with* $P(A) > 0$*,*

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\Sigma_{i=1}^{n} P(A|B_i)P(B_i)}$$

Assuming the "Naive" assumption that the features (here words of a given title) of the model are independent, and the fact that our classes (here Post Types) partition our sample space one can use Bayes theorem to calculate the probability of a title belonging to each class. The higher one is the prediction of the algorithm.

3.2 Results

The files containing the results are provided as requested in the project description, with a diagram shown after running the algorithm and acuuracy reported in terminal.

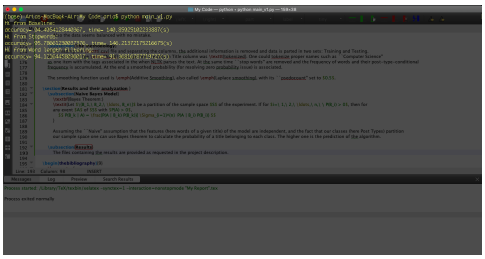

Figure 1: Example of the terminal report

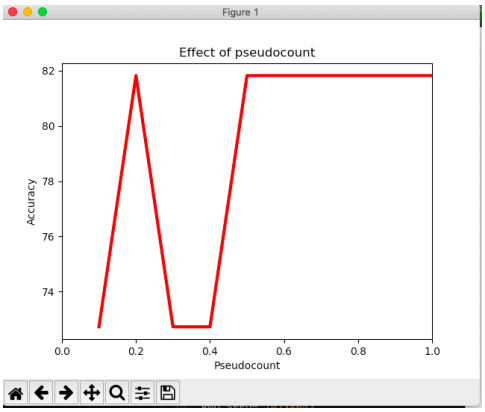

Figure 2: Example of the shown diagram

References

[1]