Spoken and Natural Language Understanding Practical Assignment 2

Submitted by Jiahui Dai

April 16, 2025

Accompanying material: Assignment_2.ipynb

1 SMS Spam Detection

Naive Bayes Classifier is used to detect spam in this task [**raschka2014naive**].

1.1 Data Preprocessing Steps

The **bag of words** model is used. It treats each document as a collection of words, disregarding grammar and word order but keeping track of word frequency.

Steps:

1. Tokenisation: breaking down a text corpus into individual tokens
2. Stop words removal: to remove words that are relatively common and uninformative
3. Stemming and Lemmatization:
   (a) Stemming: the transformation a word into its root form
   (b) Lemmatization: to obtain the grammatically correct forms of words
4. N-grams: to group a sequence of n-words together. Unigram model is performed here.

1.2 Experimental Design and Methods

1. Split the data into training and test data
2. Train the Naive Bayes Classifier model on train data
3. Test the model with train data by calculating the probability of spam/ham on a given text, and making a decision if the text is spam or ham.

   To calculate the probability of spam/ham on a given text:

$$P(X, w_j) = \prod_{i=0}^{m} P(x_i|w_j)^b \cdot (1 - P(x_i|w_j))^{(1-b)} \tag{1}$$

$$= 2^{\sum_{i=0}^{m}[b \cdot \log_2 P(x_i|w_j) + (1-b) \cdot \log_2(1-P(x_i|w_j))]} \tag{2}$$

$$\text{with } \hat{P}(x_i, w_j) = \frac{df_{x_i,y} + \alpha}{df_y + 2\alpha} \tag{3}$$

   with

   - $b \in (0, 1)$ corresponding to elements in w_j
   - $w_j \in \{\text{ham}, \text{spam}\}$
   - $df_{x_i,y}$: the number of documents in the training dataset that contains the feature x_i and belongs to class w_j
   - df_y: number of documents in the training dataset that belong to class w_j

- α: parameters of Laplace smoothing

To make a decision:

$$\text{Decision}(X) = \begin{cases} \text{spam} & \text{if } P(w = \text{spam} \mid X) \geq P(w = \text{ham} \mid X) \\ \text{ham} & \text{otherwise} \end{cases} \tag{4}$$

where

$$P(w = \text{spam} \mid X) = \frac{P(X|\text{spam}) \cdot P(\text{spam})}{P(X)} \tag{5}$$

$$P(w = \text{ham} \mid X) = \frac{P(X|\text{ham}) \cdot P(\text{spam})}{P(X)} \tag{6}$$

$$\hat{P}(\text{spam}) = \frac{\text{\# of spam messages in training data}}{\text{\# of all messages in training data}} \tag{7}$$

$$\hat{P}(\text{ham}) = 1 - \hat{P}(\text{spam}) \tag{8}$$

$$P(X) = \sum_j P(X|w_j) \cdot P(w_j) \tag{9}$$

$$= P(X|\text{spam}) \cdot P(\text{spam}) + P(X|\text{ham}) \cdot P(\text{ham}) \tag{10}$$

4. Evaluate the model with its evaluation metrics.

1.3 Hyperparameters

* Laplace smoothing * n-grams

1.4 Evaluation Metric [google_accuracy]

Accuracy is the proportion of all classifications that were correct, whether positive or negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

Recall or **true positive rate (TPR)** is the proportion of all actual positives that were classified correctly as positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

False positive rate (FPR) is the proportion of all actual negatives that were classified incorrectly as positives

$$\text{FPR} = \frac{FP}{FP + TN} \tag{13}$$

Precision is the proportion of all the model's positive classifications that are actually positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

1.5 Findings

1.5.1 TF-IDF [learndatasci_tfidf]

TF-IDF (Term Frequency-Inverse Document Frequency) is used to evaluate how important a word is to a document in a collection or corpus. It helps to weigh terms based on their frequency within a document and their rarity across all documents.

$$TF(t, d) = \frac{\text{\# terms of } t \text{ appears in document } d}{\text{Total \# of terms in document } d} \tag{15}$$

$$IDF(t) = \log\left(\frac{\text{Total \# of documents}}{\text{\# of documents containing } t}\right) \tag{16}$$

$$TF - IDF(t, d) = TF(t, d) \cdot IDF(t) \tag{17}$$

Term Frequency (TF) measures how frequently a term occurs in a document.

Inverse Document Frequency (IDF) measures the importance of a term across all documents in the corpus.

TF-IDF is the product of TF and IDF. It means that

- If a term appears frequently in a document but also in many documents, the TF will be high but the IDF will be low, lowering the overall score.
- If a term appears frequently in one document but is rare across all documents, the score will be high, emphasizing the importance of that term for the document.

1.6 Drawbacks

1.7 Potential Improvements

2 Search Engine

3 Additional Experiments