tags: IRTM

HW3 Report

Author

B06705023 資管四 邱廷翔

Environment

- python >= 3.6
- Linux >= 16.04

Requirments

- nltk

To install the required libraries, run the following command.

```
pip install -r requirements.txt
```

Executing the code

```
python main.py
```

- Before running the code, make sure the directory *IRTM* is present.

Program descriptions

The program can be broke into several phases.

1. Traverse the folder
    1. Use *tokenization*, which is from HW1, to preprocess each line of the document for the whole document collection.
    2. Form the training set, and testing set by reading the *category.txt*.
2. Feature selection
    - We will be using modified Chi-square feature selection method as mentioned in the lecture slide. The modification I made to the model is that I used tf-idf score, instead of frequency count
    - $$score_t = \sum_c \frac{N^*_{tc} - E^*_{tc}}{E^*_{tc}}, \quad \forall t \in [1, V]$$
      $$N^* = df_t \times \log(\frac{N}{df_t}),$$
    - Sort the terms by their score from highest to the lowest, and select the top **120** terms/features.
    - The selected terms will be kept and used in later training and testing.
3. Training part
    1. For each term in a category's training document:
        1. Count its document frequency in $count_t$.

    2. The Multinomial model for naive bayes can then be generated by
    3. $condProb[t][c] = \frac{count_t+1}{|V_c|+featureSize}$
4. Testing part
    1. Generate the tokenized list for each document.
    2. For each term:
        1. Ignore it if it is not in the *selected_terms*.
        2. Calculate its per class score by $score_c = \sum_{t \in V_k} \log(condProb[t][c])$
        3. Select the largest $score$ as its category prediction.