Project
Machine Learning and Deep Learning

Nguyen Thi Ly Linh (M22.ICT.003)

February 7th 2024

1 Introduction

Named entity recognition (NER) is a crucial task within Natural Language Processing (NLP), focusing on identifying and categorizing entities like people, places, organizations, and dates. This process is integral to various NLP applications, including sorting news content, automating customer support, analyzing historical texts, extracting medical information, assessing risks in finance, and aiding research.

However, NER faces several challenges such as ambiguous entity meanings, data annotation quality and domain-specific terminology. To address this, the IOB format (Inside, Outside, Beginning) is commonly used. In this tagging format, 'I-' denotes being inside a chunk, 'O' signifies no chunk association, and 'B-' denotes the beginning of a chunk directly following another chunk without interstitial 'O' tags.

In this project, I implemented Multinomial Naive Bayes with and without tokenize to perform NER.

The source code and report are defined in the following repository:
https://github.com/lylinh/NLP_Project.git

2 Dataset

The CoNLL-2003 dataset serves as a standard reference in NLP for tasks like NER and part-of-speech tagging. It encompasses entities categorized into four types: individuals (PER), organizations (ORG), places (LOC), and miscellaneous names (MISC). This dataset comprises nine classes, including B-MISC, I-MISC, B-LOC, I-LOC, B-ORG, I-ORG, B-PER, and I-PER, providing detailed labeling for entities within the text.

3 Solution

Initially, to prepare the dataset for analysis, we apply several methods:

- Eliminate any redundant characters such as ";;;" or whitespace
- Combine all words into a single sentence separated by "."
- Tokenize data

Nextly, we used Multinomial Naive Bayes with TfidfVectorizer to train. The combination is effective for text classification tasks because it leverages the power of both algorithms. TfidfVectorizer helps in representing text data in a format suitable for MNB, capturing the importance of terms in documents while reducing the impact of common words that may not be informative.

- Text data is first transformed into TF-IDF feature vectors using TfidfVectorizer.
- These TF-IDF vectors are then fed into the MNB classifier for training and prediction.
- The MNB classifier learns the probability distribution of TF-IDF features for each class during training and uses this information to classify new documents based on their TF-IDF representations.

4 Result analysis

Hence, there exists a disparity between the CoNLL-2003 validation dataset and the public leaderboard. The public leaderboard score stands at 0.52, whereas it reaches 0.93 in the CoNLL-2003 validation. This discrepancy arises from the imbalance in scores across different classes, notably evident in the considerably lower scores observed in the 4th, 6th, 7th, and 8th classes. The result of each model is presented in the following tables.

Class	Precision	Recall	F1
0	0.9558	0.9955	0.9752
1	0.8477	0.7021	0.7681
2	0.8552	0.7066	0.7738
3	0.7778	0.6865	0.7293
4	0.6960	0.4460	0.5436
5	0.8406	0.7753	0.8066
6	0.7406	0.4220	0.5377
7	0.5806	0.3186	0.4114
8	0.7733	0.5800	0.6629

Table 1: The training result by Multinomial Naive Bayes without tokenize

Class	Precision	Recall	F1
0	0.9518	0.9975	0.9741
1	0.8568	0.6748	0.7550
2	0.8758	0.7112	0.7850
3	0.8049	0.6960	0.7465
4	0.7105	0.4789	0.5722
5	0.8866	0.8177	0.8508
6	0.7977	0.3740	0.5092
7	0.6957	0.2500	0.3678
8	0.7831	0.5508	0.6468

Table 2: The training result by Multinomial Naive Bayes with tokenize

5 Discussion and future work

5.1 Discussion

After implementing several models, I have the following conclusion:

- The training time with tokenize is longer than without tokenize. However, it has better result on validation.
- Each method has different efficiency. Depending on our purposes, our requirements we can choose appropriate models.

5.2 Future work

Multiple concepts are being considered for better and potential exploration in upcoming trials and experiments. We may explore alternative training approaches, such as employing GridSearchCV or LogisticRegression.