Middle East Technical University Department of Computer Engineering

CENG 463

Introduction to Natural Language Processing

Fall 2020-2021

Assignment 1

Due date: December 15 2020, Tuesday, 23:55

1 Objectives

In this assignment, you are expected to implement a text classifier for English book descriptions. You will build classifiers using machine learning techniques on the data handed to you. Your classifiers will try to assign correct genres to book entities (a title along with a book description). Additionally, you will prepare a report to explain your models and evaluate their performances.

Keywords: *Text Classification, Machine Learning*

2 Data

The data handed to you is gathered from `https://www.goodreads.com`'s genre-based listings. The genres that we are going to consider are: philosophy, romance, science-fiction, horror, science, religion, mystery, and sports.

In each `genre.txt` file you are given the titles and descriptions of the books enlisted in that genre. In the `txts`, book titles appear at even-numbered lines and the respective book descriptions appear at odd-numbered lines. The extracted data may not be perfect. Try to get rid of any kind of noise that you think will affect your classifier's performances. For example, you might want to discard numbers appearing in the descriptions in the assumption that numbers are not going to be helpful in discriminating between the genres. Or you might want to get rid of prepositions like "by, of, from", etc.

You should split your data into 3 parts: *train*, *dev*, and *test*. The file structure that you are going to use is up to your decision.

3 Classification

The goal of text classification is to find the category or the topic of a text. Text categorization has popular applications in daily life such as email routing, spam detection, language identification, audience detection or genre detection and has major part in information retrieval tasks.

Supervised learning is the task of extracting a model out of labelled data. Each training input is assigned with a category and a machine learning algorithm uses the features extracted from the training data to build a classifier.

Processing data to extract features and selecting better ones out of many features are important tasks that directly affect the success of the classifiers. Search for text processing and representation in statistical learning tasks.

Domain is another important aspect. You are classifying book descriptions in English. Performing a literature survey might help you build better classifiers.

4 Specifications

4.1 The Classifiers

1. Name your classifier implementation module as **classifier.py**.
2. Implement a **Classifier** class with 4 methods called **train**, **test**, **save** and **load**.
3. **train** method uses the training segment of the data to learn a successful model.
4. You will use **two** classifiers chosen from the available ones in NLTK or scikit-learn. Try to adjust their parameters for a better fit.
5. **test** is used for the final evaluation of your models. It returns the accuracies of all four classifiers on test data.
6. **save** saves the trained model so that it is available for later use.
7. **load** loads a previously saved model.
8. Feel free to add any other methods such as helpers that print scores and confusion matrices, predictors that takes single files or text as their arguments etc.s
9. A template for **classifier** module:

```
class Classifier ():
    def __init__ (self):
        self.cls1 = None # Replace with a classifier
        self.cls2 = None # Replace with a classifier

        # initialization of other class members, vectorizers etc.
    def train (self, filename):
        # the training code goes here
        return

    def test (self, filename):
        # the test code goes here
        a1 = 0.0 # Replace with the accuracy of cls1
        a2 = 0.0 # Replace with the accuracy of cls2

        return [a1, a2]

    def save (self, filename):
        # save trained classifiers
        return

    def load (self, filename):
        # load previously saved model
        return

    # Other methods
```

4.2 The Report

In your reports explain the following:

1. General questions

   Precision, recall, F-score, accuracy scores,

   Micro and macro averaging of these scores,

   Confusion matrices,

   Cross validation,

   Term frequency - inverse document frequency, etc.

2. Your implementation

   Representation of data: preprocessing, cleaning, tokenization, any kind of tagging, analysis, stemming, etc.

   Classifiers: which are chosen, what are the parameters, etc.

3. Analysis of errors with confusion matrices

   These kinds of books cannot be classified correctly because ...

   These genres are difficult to distinguish because of ...

5 Regulations

1. **Programming Language:** You will use Python3.
2. **Late Submission** is not allowed.
3. **Cheating:** We have a zero tolerance policy for cheating. In case of a cheating event, all parts involved (source(s) and receiver(s)) get zero. People involved in cheating will be punished according to the university regulations. Remember that students of this course are bounded to the code of honour and its violation is subject to severe punishment.
4. **Newsgroup:** `odtuclass`

6 Submission

- Submission will be done via `odtuclass`.
- Create a `tar.gz` file named `assgn1.tar.gz` that contains all your files related to the assignment.

7 References

- NLTK Classifiers:

  `https://www.nltk.org/howto/classify.html`

  `https://www.nltk.org/book_1ed/ch06.html`

- scikit-learn:

`https://scikit-learn.org/stable/index.html`

- Information Retrieval by Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze:

  `https://nlp.stanford.edu/IR-book/html/htmledition/irbook.html`