Middle East Technical University — Department of Computer Engineering

CENG 463

Introduction to Natural Language Processing

Fall 2023-2024

Assignment 1

Regulations

1. The homework is due by **23:55 on December 13th, 2023**. Late submission is not allowed.
2. The homework consists of **2 independent tasks**.
3. Submissions are to be made via ODTUClass, do not send your homework via e-mail.
4. You can use any typesetting tool (LaTex, Word, etc.) while writing your report (but no handwriting). However, you **must** upload the report as a **searchable pdf file**. Other formats will not be considered for grading.
5. You **must** write your codes in **Python3**.
6. Upload 4 files: Your code for Task1 , your code for Task2 with feature set 2, your code for Task2 with feature set 3, your report (a single report file consisting of reports for Task1 and Task2).
7. Name your files as in the following:
   yourStudentId_Task1.py,
   yourStudentId_Task2_set2.py,
   yourStudentId_Task2_set3.py,
   yourStudentId_Report.pdf
   (e.g. 1234567_Task1.py, 1234567_Task2_set2.py, 1234567_Task2_set3.py, 1234567_Report.pdf).
   Submissions that violate the naming convention will incur a grade reduction of 10 points.
8. Send an e-mail to **garipler@metu.edu.tr** if you need to get in contact.
9. **This is an individual homework, which means you have to answer the questions on your own. Any contrary case including but not limited to getting help from automated tools, sharing your answers with each other, extensive collaboration etc. will be considered as cheating and university regulations about cheating will be applied.**

Task 1 (70pts)

You are given a textual dataset consisting of titles and descriptions of books from 8 different genres (philosophy, science-fiction, romance, horror, science, religion, mystery, sports). Making use of the given data, you are expected to implement a text classifier for English book descriptions. You will train two different machine learning classifiers: a Naïve Bayes Classifier and a Support Vector Classifier. Then, you will evaluate and compare the performance of your classifiers.

In order to simplify your job, the dataset is already divided into 3 parts: train(ing), dev(elopement) and test. Moreover, a code template is provided. You will first apply basic text processing steps that (you think) are required/effective for the given classification problem. Then you will select features (that classifiers will take into account). Thus, you will be able to train your classifiers. Classifiers will be trained on the training set. In order to fine-tune your classifiers by selecting more effective pre-processing steps and a better set of features etc., you will observe their performances on the development set. Once final versions of your classifiers are ready, you will evaluate them on the test set.

In your reports, you are expected to

1. Give the details of your implementation.
2. Explain basic text processing operations you applied. Why do you think they are required/effective/beneficial for the given task?
3. Explain your feature selection. Why do you think your features are required/effective/beneficial for the given task?
4. Evaluate performance of your classifiers. Give their **accuracies, recalls, precisions, and F1-scores** on each genre and on the overall test set. Comment on those.
5. Compare performances of NB and SVC classifiers. Did one of them significantly outperform the other? Why? Is this result expected?
6. Give the confusion matrix for your Naïve Bayes classifier. Analyse errors through the confusion matrix (These kinds of books cannot be classified correctly because ... These genres are difficult to distinguish because ...)

Additional Notes:

1. In data files, each book occupies 2 lines. Titles are at odd numbered lines, whereas descriptions are at even numbered lines.
2. For this task, you may use Google Colab environment (i.e. you may submit .ipynb files instead of .py files).

Task 2 (30pts)

You are provided formatted data (in conll-u format) and a complete, ready-to-run, Python script (uses Conditional Random Fields) for the POS tagging task. Read and try to understand the data and the code. Current implementation makes use of a simple feature set, say Set1 (Set1=$\{token, start\_with\_capital, has\_capitals\_inside, is\_all\_capitals, has\_numbers\}$). You will design 2 additional feature sets (Set2 and Set3) and compare performances of 3 feature sets. In order to be able to extract your new features, you will probably have to add a few lines to the already existing code. However, except the $extract\_features()$ function, its helper, *creatdict*() function and the main, do not make major changes in the given code (some other small changes may be required.). You may add helpers. To be more specific about what to submit and report:

1. Skimming through the given data, you will see that it consists of three columns. First column is tokens of the sentence. What are the other two columns? Which columns do we use in this task? (Hint: Taking a look into the given code and making a short survey may help.)
2. Run $Task2.py$ and report the accuracy, precision, recall, F1 values (of Set1). Create 2 new feature sets such that they outperform the given Set1. What are your new feature sets? Report. Also report their accuracies, recalls, precisions and F1 scores. Submit your codes using Set2 and Set3 (as two separate Python scripts named $yourStudentId\_Task2\_set2.py$ and $yourStudentId\_Task2\_set3.py$).
3. How would the performances of our pos-taggers affected if they have used Logistic Regression instead of Conditional Random Fields (assuming other factors such as features etc. left unchanged). Which one (i.e CRF or LR) is more suitable for this task? Why? Report.
   (Hint1: Again, a short survey may help.)
   (Hint2: Note that at the training phase classifiers are fitted sentence by sentence.)

Remarks

- You can ask anything about the assignment via e-mail or discussion forum.
- You are encouraged to use the discussion forum.
- Submit your code and report even if you cannot go far. You can get partial credit from even preliminary stages of implementation or unapplied ideas.