HW02 – ISIS 4221

Natural Language Processing 2021-I

Due date: 30-03-2021

Groups are allowed up to a maximum of 3 students or 4 only if they are the same project group. Individual work is also allowed.

Coding rules: Use jupyter notebooks and be sure that the notebook is executed and contain the results before submitting. All classes, methods, functions and free-code MUST contains docstrings with a detail explanation. Build a notebook for each point.

Report: Together with the notebooks, you must submit a written report (please use pdf format) with the answers to the questions and a short summary of the implementation.

Submission: Assignments are submitted via Brightspace. Do not email us your assignments. Please upload all files and documents.

Datasets

- **20N: 20Newsgroups (http://qwone.com/~jason/20Newsgroups/)**
- **BAC: The Blog Authorship Corpus (https://u.cs.biu.ac.il/~koppel/BlogCorpus.htm)}**
- **Multi-Domain Sentiment Dataset (http://www.cs.jhu.edu/~mdredze/datasets/sentiment/)**
- **Mourning Tweets – fnmourning.csv file.**

PLEASE READ DATASET DESCRIPTIONS

You can download all datasets from:

https://www.dropbox.com/sh/b8xuijjzmxlp51e/AAAjduNFvHmX-N16QAG39bE-a?dl=0

[25p] N-Gram Language Models Implementation

For the 20N and BAC datasets, perform the processing required to build two N-Gram Language Models:

I. Read the files and build two large consolidate files that are the union of all the documents in 20N and BAC.
II. Tokenize by sentence.
    - Normalize, but DO NOT eliminate stop words.
    - Replace numbers with a token named NUM.
    - Add sentence start and end tags <s></s>.
    - Tokens with unit frequency should be modeled as <UNK>.
III. Select 80% of the resulting sentences -random without replacement- to build the N-gram model and the remaining 10% for evaluation. Create the following files:
    - **20N_<group_code>_training (training sentences)**

- **20N_<group_code>_testing (testing sentences)**
- **BAC_<group_code>_training (training sentences)**
- **BAC_<group_code>_testing (testing sentences)**

IV. Build the following N-gram models using Laplace smoothing and generate an output file for each one (you choose the output structure, but be sure to provide an appropriate python reading method/function):
- **20N_<group_code>_unigrams**
- **20N_<group_code>_bigrams**
- **20N_<group_code>_trigrams**
- **BAC_<group_code>_unigrams**
- **BAC_<group_code>_bigrams**
- **BAC_<group_code>_trigrams**

V. Using the test dataset, calculate the perplexity of each of the language models.

VI. Create a new n-gram model using linear interpolation, define a strategy to identify a "good" combination of lambda parameters, and explain it in detail.
- Evaluate the perplexity again and build a comparative table with the results obtained.

VII. Using your best language model, build a method/function that automatically generates sentences by receiving the first word of a sentence as input. Take different tests and document them.

[25p] Naive Bayes (NB), Logistic Regression (LR)

I. You can use existing implementations of NB and LR, as well as evaluation metrics.

II. For the 20N dataset compare two classifiers NB and LR to identify the 20 different newsgroups.
- Create your own processing pipeline for the task and justify it.
- Divide the dataset into training (60%), development (10%) and test (30%).
- Train NB and LR using the following representations:
  - BOW
  - Boolean BOW.
  - Personalized representation. You as a designer must define the select set of characteristics. Explain your feature selection strategy in detail.

III. Compare the results of NB and LR using 10-fold cross validation:
- Use for cross validation: training+development sets.
- Do a search for LR hyperparameters.
- Report the average results, precision, and recall by class.

IV. Evaluate models using the test set:
- Precision, recall, F1, accuracy with the micro and macro averaging strategies.

V. Compare the results in terms of:
- NB vs LR.
- Features.
- Dataset and clasess distribution.

[25p] Sentiment Analysis

I. Use "Multi-Domain Sentiment Dataset" to build a sentiment classifier (positive/negative) per each category ("Books", "DVD", "Electronics", "Kitchen").
- Use negative.review+positive.review as training+dev dataset.

- Use unlabeleded.review as testing datasets.
- Report the results using NB as LR as classification algorithms over the test set, using as evaluation metrics precision, recall, F1, and accuracy. Use the following features representation strategies:
  - BOW
  - Boolean BOW.
  - Features only extracted from lexicons[1]. Please document which features you used in detail.
- Compare and analyze results in terms of:
  - NB vs LR
  - Features representation.
  - ***Categories ("Books", "DVD", "Electronics", "Kitchen"). Which category is more difficult to predict sentiment?, why?***
  - According to LR parameters what are the most important features per category?

II. Repeat the process but instead of building a classifier per category, build a single multiclass classifier.
- Merge all categories and build a consolidate training+dev, and testing dataset.
- Report the results using NB as LR as classification algorithms over the test set, using as evaluation precision, recall, and F1. Use the following features representation strategies:
  - BOW
  - Boolean BOW.
  - Features only extracted from lexicons.
- Compare results in terms of:
  - NB vs LR
  - Features representation.
  - **One vs multiple classifiers. Is it worth building a classifier for each category? justify your answer.**
  - According to LR parameters what are the most important features? compare with those obtained in I.

III. Investigate about decision trees (DT) and random forest (RF) classification techniques.
- What is a decision tree, how it works?
- Which algorithm is used to build a decision tree?
- Decision trees is a generative or discriminative model? justify your answer.
- What is an ensembled method in machine learning, what is bagging?
- What is random forest? How it works?.
- Random forest uses a bagging strategy? justify your answer.

IV. Repeat point II using decision trees and random forest. Remember to probe different hyperparameters for random forest model.

[1] In the dropbox link you can find some English lexicons. You are free to use any lexicon (there are many) and use them to create other features.

[25p] Mourning tweets

This dataset contains tweets regarding COVID in Spanish and English. The tag specifies whether the tweet refers to mourning or not. The EMOTICON column specifies whether the tweet contains emoticons or not. Using this dataset:

I. Build a mourning lexicon for Spanish and English.
   a. Process the text in an intelligent way, remember tweets are something noise. Use appropriate tokenization and normalization steps.
   b. Use a scaled likelihood to estimate the likely of each **word/emoticon** to appear in mourning or no mourning class.
   c. What are the most important **words/emoticons** that describe mourning? Are there differences between English and Spanish?. Build a set of visualizations to answer this question.
   d. Generate a list of mourning words and mourning emoticons with an associated score. Investigate or propose an appropriate strategy to build these lists.

II. Build a classifier to detect mourning tweets in English/Spanish.
   a. Propose a feature representation with and without emoticons.
   b. Divide the dataset for training/testing. Note that this dataset is small.
   c. Report results using precision, recall and F1 as evaluation metrics. Compare and analyze results in terms of:
      i. NB vs LR vs DT vs RF
      ii. Include or not emoticons as features.

III. Feature importance analysis:
   a. According to LR parameters what are the most important features (words/emoticons)?
   b. The “mean decrease impurity” is a method to estimate the feature importance in a RF model. Investigate this method and use it to estimate the most important features (words/emoticons).
   c. Compare and analyze the results obtained in IIIa, III.b y I.c.