Natural Language Processing (CS535)

Assignment 1

Sarcasm Detection

Question 1

Build a sentiment classifier (sarcasm detector) using bag of words model.
You can use scikit-learn (machine larning tool for python) for using implementations of classification algorithms.

Perform sarcasm classification on attached dataset. Perform 2 types of classification, binary and multiclass. For binary class, you need to classify if a tweet is sarcastic or not. For multi class you have to classify the sarcastic tweet into one of the five categories (irony, satire, understatement, overstatement, and rhetorical question). The given datasets is labelled. For multiclass, you will only use the tweets that have label of sarcastic as 1.

Split the data into train and test set by using "train_test_split(DataSet)" of scikit.

Implement following feature extraction methods.

1. Bag of words based on raw counts
2. Bag of words based on TfIDF
3. ngrams (bigrams, trigrams)

Read following links about using Vectorizer (Bag of words based on raw counts) and transformer (Bag of words based on TfIDF) for converting list of sentences to vectors

1. https://scikit-learn.org/stable/modules/feature_extraction.html
2. https://scikit-learn.org/stable/auto_examples/text/plot_document_classification_20newsgroups.html#sphx-glr-auto-examples-text-plot-document-classification-20newsgroups-py

You can use scikit learn for implementation of different classifiers as explained in above links. Use following classifiers: Naïve Bayes, Logistic Regression, Random Forest, SVM, Perceptron.

Calculate accuracy, Precision, Recall and F-score for all classifiers and report the results in tables. Make separate tables for binary and multiclass classification. For multiclass classification, report both micro average and macro average of all measures.

Question 2

Implement following sequence based deep learning models for the same task of sarcasm detection as given in Question 1. Use same dataset. Perform binary and multiclass classification. You can implement these models in Keras or Pytorch. Split the data into train and test set. Use 75% for training and 25% for testing.

RNN
GRU
LSTM
BiLSTM

For each of these models, use following hyper parameters and report the results.
Number of layers 1, 2, 3. Dropout rate, 0.2, 0.4, 0.6, 0.8
So you will have 3 *4 = 12 different sets of parameters. Report the results for each parameter setting in a table for each classifier.

Calculate accuracy, Precision, Recall and F-score for all classifiers and report the results in tables. Make separate tables for binary and multiclass classification. For multiclass classification, report both micro average and macro average of all measures.

Submission

Submit the code files and result tables as zip file on Google classroom.