95-891 Introduction to AI Homework 4: Natural Language Processing
Fall 2022

Due 11:59 PM EST March 31, 2022

Sentiment Classification

Background

In this homework, we will be working on a subset of a dataset published in the ACL (Association for Computational Linguistics) 2019 conference. The task that the dataset addresses is of generating empathetic responses to a user utterance prompt by intelligent conversational agents; e.g. when somebody says "I finally got that promotion at work," a response like "Congratulations, that's great!" is more empathetic than saying "I can't believe anybody promoted you."

The full paper is available at https://arxiv.org/pdf/1811.00207.pdf .

Goal

This homework focuses on sentiment classification. The task is to predict the class of sentiments from a predefined list of emotions.

The data set originally contained 25K snippets of personal dialogue, labeled by 32 different emotions, which we will subset to contain only 4 emotions.

You can download the dataset from:
https://dl.fbaipublicfiles.com/parlai/empatheticdialogues/empatheticdialogues.tar.gz

The subset of emotions which you will classify is {'sad', 'jealous', 'joyful', 'terrified'}

Be aware that the dataset follows a format which is given by the first line in each csv file of the dataset. We will treat attribute 'utterance' as our training attribute and the labels will be the attribute 'context'.

Note that the dataset contains a train, dev and test. You have to report your accuracies for test data wherever asked unless specified otherwise. Feel free to train on both train and dev data.

We have prepared some skeleton code to get you started, and pointed out some helpful technical blogs/material you may refer for the homework. **The skeleton code are just some helpful pointers---you do NOT need to follow it**.

In general, this homework involves two parts:

1. how to encode raw words into ML model digestible format, i.e., numerical matrix
2. after the encoding, the problem reduces to a simple multiclassification problem, from text to one of the four sentiment types.

The hw asks you to do four encodings:

1. use simple bag of words
2. use TFIDF
3. use pre-trained word2vec
4. use pretrained distilled BERT

For the first two, you could follow this blog: https://maelfabien.github.io/machinelearning/NLP_2/#2-bow-in-sk-learn (Links to an external site.)
For word2vec, you may refer to this article: https://towardsdatascience.com/using-word2vec-to-analyze-news-headlines-and-predict-article-success-cdeda5f14751 (Links to an external site.)
For pretrained BERT, you may refer to this: https://huggingface.co/distilbert-base-uncased

Now that you have experience with classification in Homework 2, here we will be focusing on building relevant features from text input. The traditional classifiers we looked at earlier work well on data which contains numerical or categorical data. In this assignment we will be exploring more on how we can represent textual data in formats suitable for building classifier models.

1. (1 point) Data ETL

   The first step is to download the entire dataset which contains many columns and rows which we will not be needing.

   a. First, filter out all rows where the sentiment (‘context’) is not in the list of aforementioned list of sentiments i.e. {'sad', 'jealous', 'joyful', 'terrified'}.

      This means you will only be retaining those examples where the sentiment lies within our set of predefined sentiments.

   b. Next, synthesize your training attributes and labels i.e. ‘utterance’ as the attributes and ‘context’ as the label.

2. (2 points) Begin by converting the utterances into a sparse bag-of-words (BOW) representation.

   Recollect that BOW is essentially a matrix of width equal to the vocabulary size of the data and each row corresponding to each utterance.

   In the sparse matrix obtained, each cell value 1 should represent that word j is present in utterance i and value of 0 indicates that it is not, where j is the column index and i is the row index.

3. (1 point) What do you think might be a shortcoming of the previous representation of utterance features? You may have guessed there are many words which are not necessarily adding much value to the classifier. For instance, words like ‘the’, ‘is’, ‘and’ are not words that help us humans identify whether or not an utterance belongs to a specific sentiment class. As you would remember, these words are referred to as ‘stop words’ in the NLP domain.

   In this step, you will remove such stop words from the utterance and build the BOW features again so that your BOW representation is free of words that do not add much value to the classifier.

   Hint: Check out the NLTK library which has a precompiled list of popular stop words

Hint: You may want to extend the existing NLTK stop words list based on some data exploration or your understanding of the domain.

4. (2 points) Normalization: Another problem with the current representation is that we weigh each non-stop-word term the same. A proven way to normalize is Term Frequency - Inverse Document Frequency (TF-IDF). This should normalize frequencies in a weighted fashion to a value between 0 and 1.

   Hint: Look at how to use TfidfTransformer in Sklearn

5. (2 points) Build a SGD classifier for the utterance sentiment classification and perform error analysis on the train data. The error analysis must include the test accuracy, confusion matrix and a few misclassified examples and your thoughts on why those utterances were misclassified by the example. You should aim for at least 60% accuracy on the test set.

6. (1 point): Build a classifier using pre-trained word embeddings like word2vec or gloVe or as the feature and an MLP classifier. Report the confusion matrix, F1 score and test accuracy.

7. (2 point): Build a classifier based on BERT and MLP. Specifically, use the pretrained distilbert-base-uncased to get sentence embeddings. You are not required to fine-tune this pretrained BERT model. And then train an MLP classifier. Explain how you use the BERT output. Specifically, which token(s) output you use? Report the confusion matrix, F1 score and test accuracy.

8. (1 point) Read the paper at https://arxiv.org/pdf/1811.00207.pdf and answer the following questions:

   1) (0.5 points) What does this paper mean by "fine-tuning" results? How might you use such fine-tuning in building an empathetic chatbot?
   2) (0.5 points) What properties of the transformer architecture make it well suited for this application?
   3) (0.5 points) Explain the metrics used to evaluate performance in Table 1 (P@1,100, AVG-BLEU, and PPL).
   4) (0.5 points) Which of the metrics do you think provides the best measure of performance of empathic systems and why?
   5) (0.5 points) Based on Tables 1 and 2, and your reading of the paper, what do you think would help the system get to human-level performance?

Submission Guidelines

The submission should be a Jupyter Notebook with the name *Intro_to_AI_HW4_<Andrew id>.ipynb.* **Answers to all these questions must be included in the first cell of the notebook that you submit. Make sure you upload only your Jupyter notebook (and not a shortcut to the notebook).**

BTW, we highly recommend starting now as this hw involves many steps :)