Capstone proposal: Dialogue classification

Moritz Bach

July 24th, 2018

Domain Background

Language processing is a very interesting field of application for machine learning, for instance in the form of sentiment analysis and text classification. The problem of classifying texts correctly is very present in our everyday lives, as all incoming email are algorithmically categorized into spam or no-spam. Thus, it can mean the difference between receiving unsolicited junk mail in your inbox and having important messages gone missing, due to it being erroneously tagged as spam. A natural choice for this problem is the Naive Bayes classifier, which is based on analyzing word frequencies and using probability theory.

Much earlier than the problem of classifying spam emails there was the problem of authorship. A popular example are the so-called Federalist Papers, a collection of essays that was published in 1787 to support the ratification of the U.S. Constitution. They were written under a pseudonym by Alexander Hamilton, James Madison, and John Jay. While authorship of most of the essays was correctly assigned years later, there still remained 12 essays, of which authorship was still disputed over by scholars.

With computational power available, Mosteller and Wallace (1963) used an approach based on Bayes' theorem and found that most of the disputed letters were written by Madison. 40 years later Fung and Glenn (2003) applied support vector machines (SVM) and found similar results.

This project will also deal with text classification. Specifically, the aim will be to classify characters in dialogues from the iconic TV show "Seinfeld". The popular comedy show ran on NBC from 1989 to 1998 and has often been called "a show about nothing". Its humor regularly picked on societal conventions and featured dialogues using sarcasm, irony and plays on words as comedic elements.

Literature:

- Mosteller, Frederick, and David L. Wallace. "Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers." Journal of the American Statistical Association 58.302 (1963): 275-309. https://www.stat.cmu.edu/Exams/mosteller.pdf
- Fung, Glenn, *The disputed federalist papers: SVM feature selection via concave minimization*, New York City, ACM Press, 2003. (9 pg pdf file)

Problem Statement

The goal of the project is to correctly predict the speaker of a given dialogue line. It is thus a multi-class text classification problem.

In order to run machine learning algorithms on the data, a bag-of-words (BOW) approach will be used. This transforms the feature space from a single column with natural text into a vector of word frequencies. The BOW approach is a simplification, which removes any punctuation and word order.

As humans we are very good at understanding natural language and interpreting the meaning of semantic nuances (e.g. wording, sentence structure, punctuation). Without learning all dialogues by heart, it might be difficult for a human to classify the speaker of a line. We might need context and we would likely try to interpret moods and motives of the character.

At the same time, characters in the show Seinfeld are rather flat types, that follow their typical behavioral patterns. Surely this is embodied by the actors through the means of tonality, body language and facial expressions. However, it can be hypothesized that the very lines a character says are already typical of her. Therefore, it will be interesting to see how well a machine can classify the speaker of a line, just from learning on word frequencies.

Datasets and Inputs

Dialogue scripts from the show are provided by seinology.com and were scraped, using the code from a GitHub repository (https://github.com/amanthedorkknight/the-seinfeld-chronicles). The data is comprised of six columns (Tab. 1) and counts 54,616 rows.

Table 1: Columns of the Seinfeld dialogues data set

Column name	Data type
ID	numeric
Character	text
Dialogue	text
EpisodeNo	numeric
SEID	text
Season	numeric

In this project, only the columns 'Character' and 'Dialogue' will be used, as labels and features respectively.

Solution Statement

For this supervised learning problem, various types of classifiers will be employed, and their performance will be evaluated and compared. In particular, the classifiers will include decision tree-based classifiers, support-vector-machines (SVMs) and artificial neural networks (ANNs).

Additionally, different data pre-processing techniques will be used and their effects on prediction performance will be evaluated. The pre-processing techniques include TD-IDF, the removal of stop-words and stemming.

Benchmark Model

As a benchmark model, a Naive Bayes classifier will be used. This is one of the oldest and extensively-studied text classifiers and thus provides a great baseline.

Evaluation Metrics

The evluation metric for the learners will simply be accuracy (ratio of the number of correct classifications to the total number of classifications).

Additionally, the multi-class confusion matrix will be shown for best the classifier, in order to interpret differences among classes.

Another possible metric is training time. However, this is irrelevant here because the dataset is small and finite. That is no new episodes of the series will be written and the learners won't be implemented in a live production environment.

Project Design

The project will be done in the language Python, inside an Ipython notebook.

The first step of the project will be the pre-processing of the data. This will include the following steps:

- Dropping of the columns 'ID', 'EpisodeNo', 'SEID' and 'Season'
- The labels of lines from non-recurring characters will be changed to 'Other'.
- All stage directions will be removed (possibly through the use of regular expressions).
- All rows containing setting informations ('Character'=='[Setting') will be removed.
- Transforming the feature space from natural language to word count vectors (BOW approach)

The data will then be split into training, validation and testing sets.

The second step of the project will be exploratory data analysis. In particular, it will be examined whether or not classes are imbalanced regarding the number of lines and as well as the number of words per line (mean/variance). This is important as very large class imbalances might call for a more detailed look at classifier performance, rather than simply comparing overall accuracy.

The third step of the project will entail the set-up of a pipeline for training and evaluating the different classifiers. The library scikit-learn and its implementations of Naive Bayes models, tree-based models and SVMs will be used. An optimization of the classifier-specific hyper-parameters will be done using grid search.

The fourth step of the project will explore the use of artificial neural networks for this classification problem. It will be determined which type of network approach is most useful, either fully connected neural networks (MLPs) or convolutional neural networks (CNNs). Then, a network architecture will be created using the library keras. This will be followed by training the network and testing its performance.

The fifth step will explore the effect of further data pre-processing steps on the performance of the learning algorithms. The pre-processing techniques will include TD-IDF, removal of stop-words and stemming. All classifiers will be re-trained for this purpose.

The sixth step of the project will be a discussion of all results, answering the question "Which classifier performed best?", among others. It will also be discussed if increases in accuracy resulted in any trade-offs (like unproportionally large training time) and how the additional pre-processing steps improved the accuracy. To discuss possible differences in classifier performance between classes, a confusion matrix of the best classifier will be shown.

To conclude the project, it will assessed how well this supervised-learning problem was solved, if there were perhaps any shortcomings and what could be possible improvements.