Warsaw University of Technology

Faculty of Electronics and Information Technology

Dimitrios Georgousis

Andrea Amato

Group 12

Introduction to Artificial Intelligence

Project 7: Sentiment analysis

Preliminary report

Description

Sentiment analysis (or opinion mining) involves identifying and categorizing opinions expressed in a text to determine whether the attitude is positive, negative, or neutral. The task will involve classifying customer reviews from the Amazon Reviews dataset available at Kaggle, so we consider it to be a classification task and, more specifically, since our dataset does not include neutral reviews, it reduces to a binary classification task. This project aims to develop, compare and improve multiple machine learning models for sentiment analysis and determine the best-performing approach.

Dataset Description

The dataset is of the form: `__label__<X> <Title>: <Text>` and this pattern repeats. Labels can only be `__label__1` (negative) and `__label__2` (positive). Most of the reviews are in English, but there are a few in other languages, like Spanish.

Data Preprocessing

Since, most of the reviews present are in English, the dataset will be striped of reviews in other languages and the models will work on the English reviews. The text will be cleaned in order to prepare the reviews for training. Cleaning process involves removal of non-essential information (e.g. hashtags, links or numbers), also multiple repetitions of letters in words will be reduced and an autocorrection tool will pass over the text. Punctuation marks and most emojis will be removed from review texts. Also, words will be converted to lowercase for ease of computation.

Label Balancing

Analysing the training set gives the following information:

```
label

2    1800000
1    1800000
Name: count, dtype: int64
```

The labels are balanced, no further action needed on label balancing.

Data Split

The dataset comes split into a training (3.600.000 entries) and a test set (400.000 entries). If a k-fold approach is employed in the training of our algorithms, additional splits into the training set may be introduced.

Performance Evaluation

Metrics

The task is binary classification and appropriate evaluation methods will be used. Metrics such as:

- Accuracy: describing the number of correct predictions over all predictions
- Precision: measure of how many of the positive predictions are true positives
- Recall: measure of how many of the positive cases the classifier predicted correctly
- F1-Score: harmonic mean of precision and recall

We can use Precision, Recall and F1-Score twice, once referring to one label as positive and the other referring to the other label as positive, to draw conclusions on both labels.

The confusion matrix is another metric which may supply useful information to us.

Data Split

We may use the default data split provided by the dataset, but also depending on exact training methods a different data split might be used as well. Conclusions will be drawn from testing on all appropriate test sets.

Solution Description

Typical Algorithms for NLP – Sentiment Analysis

- Naive Bayes is a probabilistic machine learning model that is used for classification tasks. It's based on applying Bayes' theorem with strong (naive) independence assumptions between the features. In sentiment analysis, a Naive Bayes classifier would typically treat each word (or each n-gram – n words grouped together –) in the text as a separate feature and make the assumption that each word independently contributes to the sentiment.
- Random Forest is a machine learning model that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. In sentiment analysis, a Random Forest classifier would typically use bag-of-words or TF-IDF to turn the text into a set of numerical features. Each decision tree in the forest would then make a series of binary decisions based on these features to predict the sentiment.
- LSTM is a type of recurrent neural network (RNN) architecture that is designed to remember long term dependencies in sequence data. It's particularly useful for tasks involving sequential input such as text, speech, and time series data. In sentiment analysis, an LSTM would read the input sentence word by word, maintaining an internal state summarizing what it has seen so far. The final state can be used to predict the sentiment of the sentence.
- Transformers do not process the input sequentially. Instead, they use a mechanism called attention to weigh the importance of different words in the input when

making predictions. In sentiment analysis, a Transformer would look at all the words in the input sentence at once, calculate attention scores indicating how much each word should contribute to the prediction, and use these scores to predict the sentiment.

Chosen Algorithms

The algorithms to be tested are Naive Bayes Classifier, Random Forest and Long Short Term Memory Network.

There are about 1.000.000 different words in the dataset and each review is about 70 words in length (without the title). Preprocessing of data was described in previously, but will be addressed more clearly in this segment:

- For Naive Bayes and Random Forest the text needs to be converted into a numerical format these models can understand. Such techniques are Bag of Words or TF-IDF. Also, lowercasing, stopword removal and stemming (reduction of words to their root form) may be useful so as to not represent the same word in different forms.
- For LSTM tokenization, indexing, sequence encoding, padding and embedding are typically used.

Techniques mentioned above will be explained in more depth as implementation of the algorithms progresses in later stages.