Project Poster - Text Mining for AI

Group 12

Martynas Vaznonis, Luke de Jeu, Thijs Grootjans and Ardjun Jaharia

Approaches

NERC

Bag of Words

SVM

For BoW SVM models are used since this is commonly used in NLP tasks *(lecture 4 sentiment p25)*. In particular SVM has resulted in really great performances when paired with BoW according to Maynard et al chapter 3.6.2.

kNN

For BoW kNN is used since it is simple to implement. One major downside is the long training and prediction time with large datasets. The airline-tweets dataset consists of nearly 5000 instances, so we expect a long training time *(Time complexity and optimality of kNN)*. We wanted to see how kNN performs with BoW versus WE as feature representations.

Naïve Bayes

NB determines the probability that a particular text belongs to one group or another using conditional probability for every word in the text. However we used it with BoW for NERC which works on a word by word basis and so NB can only possibly correctly identify words it has seen in the training data. Therefore, NB is mainly used as a benchmark to compare the other methods with.

Word Embeddings

SVM

Similar to BoW, an SVM trained on word embeddings is expected to have good results. The smaller feature space might require the use of an SVM with an RBF kernel over a basic linear SVM, but training and optimizing such a model requires substantially more time, and we limited this project to a simple linear SVM.

kNN

kNN works well when used with word embedding since vectors of similar words are grouped together. It determines the similarity between texts using context *(source)*. But it suffers from being too inefficient in both training and prediction to be practical.

Random Forest

We chose to implement Random Forest since it is a popular method that performs well on many different problems. The term "random forest" refers to an ensemble of decision trees, where each tree is constructed using a random subset of the training data and a random subset of the features. The final classification is then determined by a majority vote of the individual tree outputs *(Tony Yiu, 2019)*.

Sentiment Analysis

SVM

SVM is a good general use algorithm that is relatively quick to train. One advantage is that using a lot of features in our dataset we don't have to rely on the kernel trick which it makes it quicker to train. It is widely used for text polarity detection (Ahmad et al).

Naïve Bayes

The Naïve Bayes classifier learns to predict the sentiment of new documents based on the frequency of occurrence of words or features in the text. The method was chosen because it is a popular and simple method used for text classification tasks such as sentiment analysis. It has no hyperparameters and thus is very easy to train and compare with other methods.

VADER

For sentiment analysis, VADER is specifically used because it is a common knowledge-based emotional valence tool *(Maynard et al chapter 7.5)* which we already used in the lab sessions. It relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores.

Topic Analysis

SVM

SVMs are very general, and so we ended up using them for every single task including topic analysis. We have already tried using an SVM for this problem in one of the labs and the results were only a little worse than the large transformer models.

Naïve Bayes

Same as with sentiment analysis, the Naïve Bayes classifier works very well on any text classification task and thus on topic analysis. It learns to predict the topic of new documents based on the frequency of occurrence of words or features in the text. Same as before, no hyperparameters makes the setup of this model a breeze.

LDA

For topic analysis we use LDA over Bert. Bert has the advantage that it leverages contextual embeddings. However, LDA is quicker to run compared to Bert which needs Google Colab. LDA uses correlations between words in many documents to find and quantify the underlying subjects *(Jelodar et al., 2019)*. We wanted to try to implement LDA ourselves *(tm-ba-lecture-6-topic-modelling p19)*.

Data usage

Collect Data

The airline-tweets dataset from Lab 3 was used for the training of the sentiment analysis. This dataset contains almost 5000 gold-labeled instances. From lab 4, the CONLL2003 dataset was used for IOB data. For the training of Topic Analysis, new datasets were gathered from Kaggle, including a dataset for books and restaurants. A dataset for movies was gathered from an NLTK module called movie_reviews.

Clean and tokenize

Stop word removal, POS-tagging and tokenization is applied using NLTK and Pandas. Irrelevant columns are dropped from the data sets. At the end lemmatization is done when using LDA for topic classification.

Transform and split

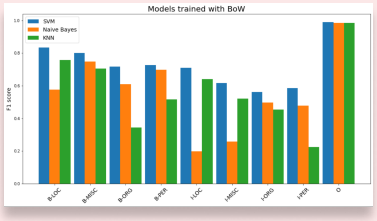
The data is transformed to the required format, including, embeddings, BoW and TF-IDF. The data is then split into a train, validation, and test sets.

Results

NERC

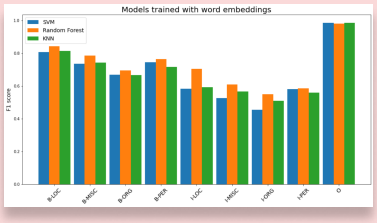
Bag of Words

Observing the results over the validation set tells us much about the performance. For NERC with BoW, looking at the validation data plot to the right, it can be seen that SVM has the highest F1-score for every label. From the classification report, we see that the Macro F1-score of SVM is 0.73, while both NB and kNN have 0.56. The accuracy and weighted average are similar to each other and all have a score between 0.91 - 0.94 with the SVM having the highest (0.94). Using SVM with BoW gives the best classification results for NERC.



Word Embeddings

Again, we can observe the results over the validation data in the plot to the right. For NERC with WE, the plot shows that Random Forest has the highest F1-scores except for the O-label. The classification report, the macro average F1 score is higher for Random Forest with around 0.72 than the others with both 0.68. The accuracy and weighted average are similar to each other and Random Forest having a slightly higher weighted average 0.94 compared to the other two having 0.93.



Sentiment Analysis

When comparing VADER, SVM and Naïve Bayes, Naïve Bayes seems to perform best. Their accuracy scores are respectively 0.60, 0.81, 0.86 when measuring the results over the validation set. The dataset is balanced when it comes to negative, positive and neutral instances. Despite its simplicity, Naïve Bayes works really well for such task and even outperforms a basic SVM. While the results form VADER were substantially worse, they still performed far better than random guessing which is remarkable given that no training was undergone.

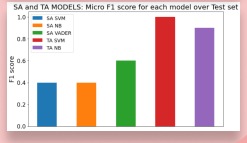
Topic Analysis

When comparing SVM, Naïve Bayes and LDA, SVM seems to perform best. SVM had very good results with an accuracy of 93%, and Naïve Bayes had an accuracy of 90% when measuring over the validation set. The topics deduced by LDA do not seem to conform to the three topics of movies, restaurants, and books and thus classification using this model seems futile.

Test Set Results

Looking at the plot for the NERC method, it can be seen that the Macro F1 score is highest for SVM with BoW, with a Macro F1-score of 0.56. All NERC models perform relatively similar, and have F1-scores in the range of 0.46 - 0.56. These scores are not very high and could be due to the small test set size. These relatively low scores could also be the result of a difference in topic between the train and test set. Additionally, there results are somewhat in line with the results of the validation set, as SVM BoW also performed the best for that test. The Macro F1-scores were used instead of the Micro F1-scores, to maintain representative results regardless of the strong class imbalance.

Looking at the plot for Sentiment Analysis and Topic Analysis, it can be seen that the VADER performs the best for SA, with a Micro F1-score of 0.6 while SVM and NB perform equally low. For all SA models, the performance is relatively low. This was likely due to the training data being of a significantly different distribution of words and topics compared to the test set.
The TA models performed quite well. The TA SVM had a perfect score, while TA NB still scored a Micro F1 above 0.85. For the SA and TA models, the Micro F1-scores were used, as there is no strong class imbalance.



Limitations

For most of the methods used, an initial grid search was run to find and tune the hyperparameters to the most effective values. For some models, this grid search had to be executed within smaller bounds and/or larger steps, due to runtime limitations. With more time and resources, each model could be trained with more optimal hyperparameters, allowing the methods to perform closer to their true potential. Additionally, testing the performance over a larger test set would have produced more accurate and nuanced performance values, compared to the small test set in the current setup.

Division of work

Ardjun: Contributed mainly to the coding of the Topic Analysis. Analyzed the Topic Analysis results and helped overall with finding relevant literature. Helped write out the description and motivation for our approaches poster text.

Luke: Contributed mainly to the coding of the NERC with embeddings. Analyzed the NERC embeddings results and generated the plot. Designed and created poster using mostly text provided by teammates. Created the division of work.

Martynas: Contributed towards the coding of the NERC with BoW, and oversaw all other programming, helping to solve issues and streamline process. Analyzed NERC BoW results and generated the plot. Contributed to poster text final cleanup.

Thijs: Contributed towards the coding of the Sentiment Analysis. Analyzed the results of the Sentiment Analysis. Helped transfer results to text suitable for the poster, and finding relevant literature.

Source code

The .zip file of the complete source code, including the used datasets and generated models can be found at: https://drive.google.com/drive/folders/1zF_4CzDFy9UJFHFwCpyjjiWSTM1DSxYz?usp=sharing