Conference handbook

Natural Language Processing (2023)

P01: Exploring BERT and VADER Ensemble Methods for Sentiment Analysis

This project investigates the possibilities of ensemble existing techniques of NLP sentiment analysis to increase performance on the task of classifying IMDb movie reviews. As a baseline to the project, we implemented the pretrained "bert-base-uncased" and fine-tuned it to our data set. As a secondary module used in the ensemble methods, we implement the rule-based sentiment analysis tool VADER. We hypothesized that by combining word specific data from a rule-based model with the data from a context based model, strengths from the respective models could complement each other, resulting in an increase in performance. The project takes inspiration from the work on VADER and BERT ensemble methods by Wang et al., but explores two simplified methods. The first method concatenates the output from VADER and BERT and further trains the weights using an MLP. In the second method, the model outputs are multiplied with different weights and then added. The weights are simply updated with a loop and are not optimal, but show an approximation of the relationship between the models. Our findings indicate that both methods have a slight positive effect on the model performance compared to the baseline, with both methods yielding an increase in the accuracy measure.

P02: Predicting Political Party Affiliations Using BERT with Adapters

We used the state-of-the-art pre-trained language model BERT with an addition called adapters to predict the political party affiliation of a speaker based on their speech. The project aimed to provide insights into the linguistic patterns and characteristics that are associated with a particular political party's ideology and beliefs. The dataset used for the project is from Riksdagens Öppna Data and contains speeches from politicians belonging to various political parties, and the model is trained to classify them into their respective parties based on the text input. The project's key contribution is the use of adapters to fine-tune the pre-trained BERT model, which together with the adapters were able to classify parties correctly with an accuracy of 59.35%. The baseline model (BERT with one classification layer) were only able to classify parties correctly with an accuracy of 34.22% clearly showing the advantage of adapters.

P03: A Look into Dependency Parsing – Evaluating Arc Transition-Based Dependency Parsing Techniques

The purpose of this project was to implement and evaluate different extensions to a fixed window prediction dependency parser. The parsers were trained and evaluated on the Universal Dependency projects English Words Treebank (EWT). The extensions we evaluated were Beam Search, Arc Hybrid parsing, using a Dynamic Oracle, and learning sequences of arc transition parsing with the help of Attention. Experiments were conducted with gold standard tags and tags predicted by a trained fixed window neural network model. We found that the Attention parser improved on the baseline noticably, while other parsers did not reach the same performance as the baseline. Our intuition gathered from previous articles was that all extensions should have improved the results, and it is possible additional tuning or refined implementations could improve the results we achieved.

P04: Dependency Parsing Using Bi-LSTM and Arc-Hybrid System

This project has aimed to improve the dependency parser which was developed in the baseline. To do this, we have implemented a bidirectional LSTM (BiLSTM), inspired by Kiperwasser and Goldberg (2016), which is trained along the fixed-window model. The baseline consists of a POS tagger and a dependency parser. The BiLSTM is trained on each sentence in the dataset. We also implemented the Arc-hybrid algorithm and compared how it affected the accuracy. The implementation was tested with the POS-tagger which was implemented in lab4 and the dataset, English Web Treebank. There we could see an increase in accuracy from 69% to 74.6% with only the BiLSTM. Meanwhile, with Arc-hybrid we got an accuracy of 75.2%. Further improvements to the model were to be done, but due to time constraints, there was no time to completely implement it. This improvement was a dynamic oracle instead of the static oracle which was used in the baseline.

P05: Entropy Regularization – A Method to Reduce Repetition in Auto-Generated Text

This project explores the use of entropy regularization in fine-tuning a GPT2 language model, inspired by the work presented in the Master's thesis *Regularized Fine-tuning Strategies for Neural Language Models* written by Jae Eun Hong. We replicated Hong's method of extending a generative language model with entropy regularization, but using the SimpleWiki dataset. In addition, we evaluated our baseline and our extended model using methods such as perplexity and per token-probability to assess the impact of entropy regularization on the performance of the language model. The project can be further extended to explore other models and datasets.

P06: Friends Are All You Need

In this project we have created an end-to-end system for answering questions about Friends, by using Sentence-BERT for automatic context retrieval and BERT-QA for formulating the answer. For this we are using the Friends dataset from the FandomWiki. This is then parsed using XML, to have separate documents for each page. For the evaluation, we handpicked 50 questions which were different in terms of syntax and information asked. We compared the performance on the evaluation questions with a baseline based on a conventional method for context retrieval called TF-IDF. From this, we saw an improvement over the baseline from 20% to 40% accuracy. This was also compared with only the QA-BERT model using the gold contexts which achieved 80% accuracy. Clearly, the retrieval part is the main limiting factor of the performance of the system. S-BERT clearly outperforms the TF-IDF thanks to a better semantic understanding of the text. Neither are very good at understanding implied information that is not explicitly stated.

P07: Comparing Sentence Parsing with Arc-Hybrid against Arc-Standard

Using the baseline developed in the lab series, we tested and compared the sentence parsing performance of the Arc-Hybrid algorithm to the Arc-Standard algorithm, using dynamic and static oracles. The metrics used to measure performance were the parsing accuracy and UAS score. The models were tested on English, French, German, Spanish, Swedish, Chinese, and Hindi language datasets. To get an accurate picture of the model performances, we tested Arc-Standard on the static oracle, and Arc-Hybrid on both static and dynamic oracles on each of the language datasets. We decided to not test the Arc-Standard algorithm using a dynamic oracle due to time constraints. From our results, it is clear to see that the Arc-Hybrid algorithm outperforms the Arc-Standard, but it is unclear whether using the dynamic or static oracle made any difference. A topic for future work would be to rerun the tests including Arc-Standard on dynamic oracles.

P08: Dependency Parser Benchmark

In this project we have extended the arc-standard and static oracle dependency parser created in Lab 4/5 with an arc-hybrid and dynamic oracle dependency parser. Furthermore, additional features for the dependency parser were implemented. The baseline is trained using arc-standard and static oracle. In this project, the baseline dependency parser is compared to two other dependency parsers created: arc-hybrid with static oracle and arc-hybrid with dynamic oracle. Implementing arc-standard with dynamic oracle is skipped, as it requires a dynamic programming algorithm and we had limited time. Moreover, these dependency parsers are both trained using part-of-speech tagged sentences from the tagger created in Lab 4, and golden part-of-speech tagged sentences given by the Universal Dependencies treebanks. We use projectivized English and Nynorsk treebanks to evaluate the dependency treebanks based on UAS and LAS. The results show that for all test cases the golden part-of-speech tags outperform our tagger which is expected. For the English treebank we see a decrease of 3% accuracy when using the arc-hybrid with dynamic-oracle over using the arc-hybrid with a static oracle, the latter giving accuracy of 73%. An interesting finding is that using the arc-hybrid with a static oracle outperforms all the other parsers.

P09: DocBERT Starts Daytrading – A Case Study on Transfer Learning for News Classification

This project investigated whether a fine-tuned BERT model could accurately predict how financial news articles affect stock prices. We used a pre-trained Swedish BERT that we fine-tuned using two different datasets: financial news articles and Amazon baby product reviews. For testing, we used our own gathered financial news articles labeled as either negative, neutral or positive. A DocBERT was implemented, which splits long documents into shorter sequences with overlaps, thus achieving linear time complexity. The project also investigated if the training data length impacts DocBERT performance by sampling a larger different dataset and creating replicas of the datasets. The results of the test revealed that DocBERT's performance did not appear to be adversely affected by document length. The model trained on translated financial phrases had a total accuracy of 52.98%, while the model trained on translated Amazon reviews had a total accuracy of 41.27%. Although the obtained accuracies were lower than expected, the project still provides evidence of the potential for natural language models and transfer learning to predict short-term changes in stock prices.

P10: Exploring Automated Essay Grading Using GPT Models

For our project, we considered the problem of automated essay scoring using GPT models. We explored the capabilities of zero shot learning on GPT-3.5-turbo, as well as fine-tuning on GPT-3-ada and GPT-3-davinci. The results were compared to a baseline system using augmented memory networks which achieved a quadratic weighted kappa score of 0.78. During our trials of fine-tuning we achieved kappa scores of 0 regardless of model variations and training time. This would be equivalent to randomly assigning grades suggesting that we made an error in the implementation. The cost of accessing the OpenAI API and the time limit prohibited us from doing further experimentation. The zero shot learning was mostly about how to do proper prompt engineering and how to evaluate the quality of the responses from GPT-3.5-turbo. After testing different structures of prompts we settled on a prompt that achieved a quadratic weighted kappa score of 0.49. Although this is not an improvement compared to the augmented memory network model, the score is significant enough to show that GPT-3.5-turbo is able to give scores comparable to teachers. However, our implementation is not mature enough to replace teachers.

P11: Explainable Bengali Multi-Class News Classification

Text classification is one of the fundamental natural language processing tasks that helps to automate the process of sentiment analysis, information retrieval, etc. Bengali is the fifth largest language in terms of native users. Though text classification is not a new research topic, there is scope for improvement in the context of the Bengali language. In this work, we use a pre-trained BERT model and fine-tune it for multiclass text classification. We use a publicly available dataset containing over 400K Bangla news articles with nine classes and achieve 93% accuracy. Additionally, we use Integrated Gradient, an explainable AI technique, to explain the outcome of our model. We show which words in a news article affect the model to choose a particular class.