Sentiment Analysis from ESG Point-of-View Using ML

Oscar Johansson, Alexander Möhle

INTRODUCTION

ESG is a concept that is used to evaluate companies based on their long term *environmental, social and governance risks*, e.g. energy efficiency, worker safety, and board independence. Since most pollution comes from a minority of companies, those who are publicly traded may curb their pollution when investors leave from a deteriorating *ESG* score.

Sanctify is a fintech company based in Lund which focuses on the development of AI-based financial analysis software, with an emphasis on *ESG*. They also provide access to their data in the form of an API, with the target audience for their applications being mainly fund managers. A part of their pipeline is the processing of a large number of news articles to determine the *ESG* scores of companies. One step in this processing is to measure the sentiment of the news articles through *sentiment analysis*.

In general, sentiment analysis means trying to determine whether a given text expresses itself as positive, neutral or negative. There exist different approaches to sentiment analysis such as *lexicographical* and different machine learning algorithms, e.g. *LSTM* or *Naive Bayes*, as well as a hybrid of the two approaches.

Sanctify's current solution is a lexicographical approach that is tuned with the addition of ESG-related terms. Modifications to the lexicon are needed to get an *ESG* perspective. For example: "Company X has increased its greenhouse gas emissions". Without an *ESG* perspective, "greenhouse gas emission" would have no meaning and the text would in the best case be classified as neutral, and worst case positive. However, even with the modifications, the lexicon-based approach is not very generic and requires care when changing the lexicon.

This thesis will evaluate a *transfomer* machine learning approach and compare it to a lexicographical approach.

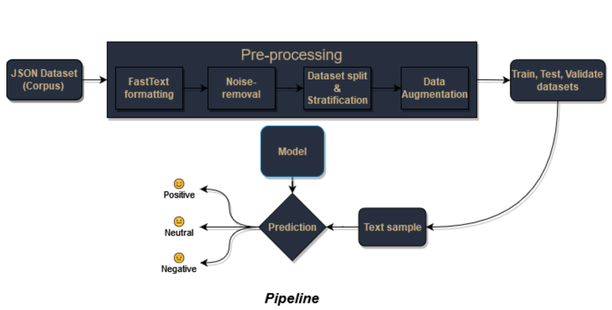
RESEARCH QUESTIONS

- How is sentimental analysis done currently?
- What tools can be used for a machine learning solution?
- How should different solutions be compared?
- How can a transformer model be optimized for text classification?
- What tools exist that can augment a ESG-based dataset for NLP?

METHODS

The main development method used to test and produce results in this thesis was prototyping. Different architectures were developed and tested in order to understand how they worked and how they performed compared to each other. Initially, commonly used methods were utilised and then, as development went on, more modern and in-depth methods were tested.

In total, this thesis developed five different pipeline variations. One for zero-rule approach, one for the lexicon approach and finally 3 for the *transformer* ML approach. The first *transformer* approach used default parameters, the second used optimized hyperparameters by running hyperparameter searches, and the final used fine-tuning.



Pipeline

An *Zero-rule* classifier was used as a comparator. Since most texts were neutral, it was used to do a majority *Zero-rule* prediction on the test set, i.e. Zero-rule classifier only classified texts as neutral.

The lexicon baseline in this thesis was made to be as close as possible to the already existing solution that *Sanctify* uses. It was, however, not a copy of their entire solution. Only the *Python* library *VADER*, which handles the classifying of the articles, was implemented.

To optimise *VADER* to the *ESG* use case, a word list was implemented. *Sanctify* aided with implementing the word list by suggesting some common *ESG*-terms to be added to thte word list.

Different *transformer* models were tested against each other to find which one performed the best. All of the models tested are under the *BERT* family tree.

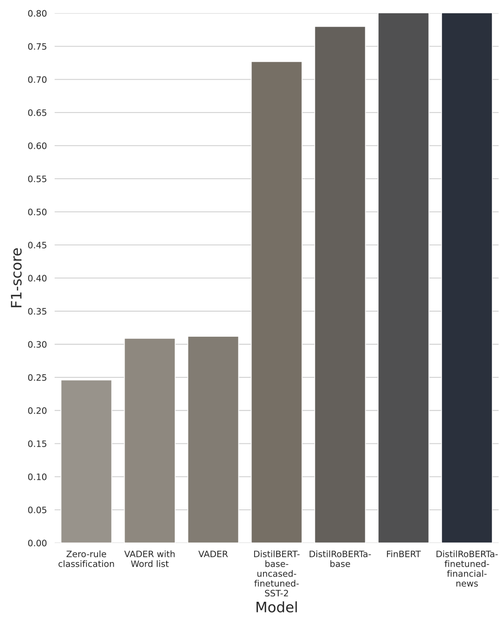
In hopes of getting the most out of the models that were chosen from the first test, all the chosen models went through a hyperparameter search.

The two models that had the best average macro average F1-score after hyperparameter seach was then chosen to be further fine-tuned.

RESULTS

Since most of the texts in the test set was neutral followed by positive and lastly negative, it meant that the test set was imbalanced. The most fair method to compare the different models would therefore be the macro average F1-score.

Name	Type	Macro avg F1-score
Zero-rule classification	Baseline	0.246
VADER with Word list	Lexicon-based	0.309
VADER	Lexicon-based	0.312
DistilBERT-base-uncased-finetuned-SST-2	Transformer	0.727
DistilRoBERTa-base	Transformer	0.780
FinBERT	Transformer	0.802
DistilRoBERTa-finetuned-financial-news	Transformer	**0.804**



CONCLUSION

In this thesis, a corpus of ESG-related news articles were used to train four transformer-based machine learning models. The objective was to predict the sentiment of the articles positive, neutral or negative with a higher accuracy than the current solution at Sanctify.

Compared to the lexicographical solutions explored in this thesis, the results show show that *BERT*-based models perform markedly better in the sentiment classification task for the *ESG* dataset used in this thesis. The average macro average F1-score of each model was evaluated after 10 runs, to establish a confidence interval for each model. From a selection of four initial models, the best performing was selected in each iteration for hyperparameter tuning and fine-tuning. The final models selected for fine-tuning were the models *DistilRoBERTa-finetuned-financial-news* and *FinBERT*. *DistilRoBERTa-finetuned-financial-news* achieved the highest macro average F1-scores of 0.804 from being trained with default hyperparameters.

A learning experience from the duration of this thesis was the realization of the minor impact hyperparameter tuning and fine-tuning had on the final results. In hindsight, it would have been more promising to explore more models in their default state, with a focus on more pre-training.

ACKNOWLEDGEMENTS

We would like to first and foremost thank Gustav Johnsson Henningsson and Henrik Ljunger for participating in our weekly meetings and providing valuable suggestions. We would extend this thanks to Patrik Elfborg for being incredibly reliable whenever we had issues with Linux and Git. Last but not least we would like to thank our supervisor Marcus Klang for providing us with lectures worth of knowledge about machine learning, and for his guiding feedback throughout this thesis.