**Plan of work (appended to thesis contract)**

Name of student Michiel van Nederpelt

Degree programme Linguistics Text Mining **student number** 2684581

Name of supervisor(s)/ Ilia Markov (VU Amsterdam, Faculty of Humanities, CLTL lab)

Second reader/ Hennie van der Vliet (VU Amsterdam, Faculty of Humanities, CLTL lab)

Chair

Working title Evaluating a transformer-based language model for in-domain and cross-domain  
offensive language detection on perturbated data

Topic Evaluation of a transformer-based language model for offensive language detection, both under in-domain and cross-domain conditions with the use of checklist for introducing perturbations. The BERT transformer-based pre-trained language model shows near state-of-the-art offensive language detection performance. Behavioural testing is used to get deeper insights in the model’s performance. By perturbation of data through checklists the model will firstly be tested under in-domain conditions, secondly the in-domain setup will be expanded by introducing perturbations, then cross-domain performance will be evaluated without perturbations and there after cross-domain performance with perturbations.

Aim and relevance The topic of offensive language detection has seen major interest not only by the natural language processing community, but rather by society as a whole. In an era of high paced information interaction and platforms like Twitter, Facebook and YouTube open for discussion and conversation for anyone with an account, the call for offensive language detection and potential blockage of users ventilating offensively has risen significantly. As offensive language detection models are improving performance people have adapted accordingly. The use of special characters, numbers and small alterations of letters in words are a way of “cheating” the detection model. By monitoring BERT performance under increasing challenging conditions the ability of the model to perform in the “real world” will be evaluated. Perturbations will be focussed on typos to get insight in the model robustness performance. Error analysis of the system will be performed both quantitative and qualitative. This project aims to give an insight in the near state-of-the-art BERT offensive language detection model performance both on in-domain and cross-domain datasets with perturbations. The goal of the thesis is to get a better understanding of how the model performs under increasingly challenging conditions. We add to the practice of challenging dataset usage for model evaluation by applying it to a cross-domain study.

Problem definition Research question: What is the performance of a transformer-based pre-trained language model (BERT) that shows near state-of-the-art results for the offensive language detection task under challenging conditions

Sub-questions:

* What are the main differences in the domains tested (topic, platform etc.)?
* What is the performance of BERT on offensive language detection for two different domains?
* What is the performance of BERT on offensive language detection under challenging conditions (“in the wild” testing)?
  + Perturbations focus on typos (robustness)
  + Cross-domain
  + Cross-domain and perturbations
* What are the insights gained from differences in performance across different setups of model training and testing?
  + Reasons for cross-domain drop
  + Reasons for drop with perturbations

Data-collection Two existing annotated datasets for offensive language detection will be used:

**OLIDv1;** Dataset used in the SemEval 2019 shared task for offensive language detection.

**HASOC**; The HASOC track intends to stimulate development in Hate Speech for Hindi, German and English. Although this study focusses on the English dataset only. Three datasets were developed from Twitter and Facebook and made available. Binary classification and more fine-grained subclasses were offered in 3 subtasks.

Research Method The following approaches will be used:

1. The language model BERT, short for Bidirectonal Encoder Representation from Transfomers, is a state of the art Machine Learning-based NLP model developed by Google. BERT is used for multiple NLP-tasks, such as offensive language detection, Neural Machine Translation, Question Answering, Sentiment Analysis and Text Summarization. BERT can be trained to understand language, which is done in the *pretraining phase*, while in the *fine tuning phase*, the learnt language is used for specific tasks. To learn language, two unsupervised tasks, Masked Language Modelling and Next Sentence Prediction are used. The BERT model to be used is already pretrained but finetuned on the task of offensive language detection.
2. Datasets (for description of datasets see “Data-collection” above)
   1. In-domain:
      1. Train OLID (train.csv) – test OLID (test.csv) + **checklist**
      2. Train HASOC – test HASOC (same label distribution as in OLID test.csv) + **checklist**
   2. Cross-domain:
      1. Train OLID (train.csv) – test HASOC (test) + **checklist**
      2. Train HASOC (train) – test OLID (test) + **checklist**
3. Error analysis
   1. Quantitative
      1. F1-score with recall and precision for overall model performance
      2. Minimal functionality test with failure rate for perturbations
   2. Qualitative
      1. In-domain misclassifications
      2. Cross-domain misclassifications
      3. In-domain vs. cross-domain misclassifications overlap and differences
      4. “Deep dive” in the failure rate for perturbations

Provisional organization of chapters

1. Abstract
2. Introduction
   1. What is offensive language
   2. Relevance of the overall task
   3. Problem definition
   4. Goal and research question
   5. Overvieq of chapters
3. Related Work
   1. Overview on previous works on offensive language detection
   2. Similar approaches
      1. Overview on previous works on similar tasks or “solved” task used by models
   3. Tasks which have been performed under the “umbrella” of offensive language
4. Methods
   1. Neural networks
   2. Transfer learning
   3. Models
      1. Evolution of models used for language detection overall
      2. Introducing/describing the transformer-based BERT model
   4. Behavioural testing
      1. Checklists
5. Data
   1. Description of dataset
   2. General statistics
   3. (Description of pre-processing steps)
   4. Domain differences
   5. Test/train distribution
6. Results
   1. Description of Results
7. Discussion
   1. What insights do the experiments provide?
   2. Quantify performance as a:
      1. Drop after introducing perturbations
      2. Drop in cross-domain
      3. Drop in perturbations in cross-domain
   3. What are the implications of in-domain vs. cross-domain? The perturbation used to check model performance and bridge to real-world platform posts
   4. Error Analysis
   5. Future Work
8. Conclusion

Provisional book list

(appendix)

Challenges for Toxic Comment Classification: An In-Depth Error Analysis – van Aken et al. arXiv:1809.07572v1

Challenges and frontiers in abusive content detection - Proceedings of the Third Workshop on Abusive Language Online, pages 80–93 – Vidgen et al.

HATECHECK: Functional Tests for Hate Speech Detection Models – Rottger et al. - arXiv:2012.15606v2

See much more in thesis

Timetable

|  |  |  |
| --- | --- | --- |
| Phase | Dates | Tasks |
| Orientation Phase | 01-04 – 10-04 | * Explore dataset and literature * Start thesis planning and finding research approach * Familiarize with Ilia as a thesis supervisor * Find working place |
| Official start | 05-04-2022 | * officially begin with the thesis project |
| Execution Phase | 01-04 – 17-4  2.5 week | * Pre-processing * Prepare datasets (merging, labelling, formatting etc.) * Start writing and structuring:   + Related Works   + Dataset description |
| 17-04 – 08-05  3 weeks | * Run classification experiment on in-domain trained classifiers * Write   + Model(s)   + Methods   + Related Works |
| 08-05 – 05-06  4 weeks | * Run classification experiments * Evaluation and Comparison of Results   + Write Results section * Start working on Error Analysis, taking notes while evaluating |
| 05-05 – 10-06  1 week | * Write Discussion   + Error Analysis   + Future Work |
| Planned thesis presentation date | 23-05 | |
| Completion phase | 10-06 – 19-06  2 weeks | * Write   + Abstract   + Introduction   + Conclusion * Clean up code |
| Final Phase | 20-06 – 29-06-2022  1.5 weeks | * Finalize everything * Incorporate last feedback * Proofread |
| Planned submission thesis | 29-06-2022 | |
| Planned graduation date |  | |

Remarks: