Literature Survey and Project Plan Group 23

Tsuheng Hsu
tsu-heng.hsu@aalto.fi

Marius Boda
marius.boda@aalto.fi

Trung Ngo
trung.ngo@aalto.fi

Contents

1 Literature Review

1.1 Task Overview

The task of our project is Multilingual Toxicity Detection, where the goal is to develop text-based models that classify short texts as toxic or non-toxic. The key challenge for the task is the generalization of cross-linguals between English, German, and Finnish, as the training dataset only contains English. In contrast, the development and test dataset contains German and Finnish.

Therefore, the main goal of the project will be then:

- Choose, apply, and compare tokenization and embedding methods that effectively generalize to English, German, and Finnish.
- Train and evaluate both custom and pretrained models for toxicity classification, and compare their effectiveness.
- Explore additional techniques to enhance classification performance for the competition.

1.2 Tokenization

Text embedding and tokenization are fundamental steps in natural language processing (NLP), which will significantly impact the performance of one's final model. The goal of tokenization is to convert raw text into smaller units such as words or subwords as the base unit of the entire model.

There are many strategies for tokenization, such as word-based tokenization which is the most intuitive way of tokenizing text by segmenting text by treating each word as a discrete unit. While this method can be effective for some languages, its performance varies depending on linguistic characteristics.

For English and German, it may be sufficient due to their analytic and fusional nature, where each words often have discrete units of meaning, but it may also fail to capture the full morphology of the entire text. In the case of Finnish, which is an agglutinative language, word-based tokenization can be particularly problematic, since a Finnish word can have numerous inflected forms e.g. "talossa" ("in the house"), "talostani" ("from my house"), "talojeni" ("of my houses"). So, subword tokenization such as Byte-Pair Encoding (BPE) [1] is a good alternative as it breaks the words into smaller, meaningful units across inflected forms, and helps improve model efficiency and generalization across morphologically rich languages like Finnish while still being effective for English and German.

1.3 Embedding

After tokenizing the text, it must be transformed into a numerical representation that can be processed by the models. Text embedding embeds words into dense vector representations

capturing the semantic and syntactic relationships. Common approaches [2][3] utilize neural network-based approaches to learn word representations by predicting word co-occurrence in large corpora. These methods may be sufficient for explicit text embedding, where words have clear and direct meanings. However, for text with implicit hints, contextual dependencies, or polysemy, static embeddings may struggle to capture the nuances of meaning since they assign fixed representations to each word regardless of context.

More recent embedding approaches such as [4] [5] introduced contextual embeddings, where word representations dynamically change based on their usage in the text, and [5] further enables cross-lingual transfer by aligning word representations across multiple languages.

1.4 Theoretical Difficulties

Key challenges include cross-lingual generalization, the complexity between languages, and how embeddings work in this case. Because most of the training data are only in English, models often have very poor performance in Finnish's very rich inflectional morphologic language. BPE has possibilities to work but may sometimes provide inconsistent results across languages. To add on to the difficulties, toxicity can be subjective. It may vary from language to languages and based on cultural contexts. Basic tokenization techniques will fail as the training data cannot possibly cover all possible words in Finnish, English, and German. This means that words that have not been seen before, and are present in the test set, will have a OOV (out of vocabulary) token assigned to them. This decreases accuracy of the test set and ultimatly makes basic tokenization not very robust in the case of multilingual toxicity detection.

1.5 Model Architectures for Text Classification

Language is inherently sequential, where the meaning of the sentence or text depend on the sequence of the words. Therefore, a model needs to be able to capture local and long dependencies of the words in a sequence. This leads to the development of various sequence-to-sequence models.

Early approaches such as Convolution Neural Networks (CNNs) [6] apply convolution filters across words to capture dependencies across them. Further developments such as Recurrent Neural Networks (RNNs) [7] which are designed especially for sequential data, process each sequence individually and maintain a hidden state that captures information from the previous sequence, allowing each sequence to be processed with the dependencies of the early sequences. And further improvements such as Long Short Term Memory (LSTM) [8], Gated Recurrent Units (GRUs) [9], and bi-direction RNNs [10] further gap the limitation of normal CNN and RNNs capacity in range and direction of capturing dependencies. However, despite the improvement the limitation of modeling global context is still limited, especially in longer sequences.

To push the limitation, attention based transformers [11] is introduced which is now the state-of-the-art architecture to process sequence data. As toxicity detection does not need a decoder, an encoder only transformer model is adequate for this task. A transformer encoder is defined as a stack of N encoder layers [12].

BERT or Bidirectional Encoder Representations from Transformers models are currently state-of-the-art models that can be fine-tuned with an additional output layer [4]. During BERT's pretraining researchers used WordPiece embeddings with a token vocabulary of around 30,000 [13]. BERT masks some words in a sentence or sequence and then forces the model, by training, to learn to predict those words by using the context on the left and right. This essentially teaches bidirectional representations. In addition, BERT utilizes transformer-based architecture., specifically the self-attention component that allows every word to attend to all other words in the sequence.

2 Project Plan

2.1 Current Progress and Findings

For our first model, we applied an encoder-only RNN model as our baseline model with Natural Language Toolkit's Word and Punctuation Tokenizer which is a word-based tokenizer, and Torch's built-in embedding functionality. We then compare it with a transformer-based model with the same tokenization and embedding with 3 attention blocks. The results of the models on the development set are shown in Table 1, which shows that the transformer-based model outperformed the baseline model. Still, as shown in the table, the model is poor in generalization across languages, especially in Finnish due to the morphology difference compared to the other two languages.

Therefore to make the model able to adapt to other languages, we further trained another encoder-only Transformer model but with BPE tokenization which aligns more with Finnish morphology. We utilized Google's SentencePiece [14] BPE tokenizer and trained it with English-Finnish corpus from Finnish Information Bank [15] and 10k German corpus from [16], and further increased the model's attention blocks to five. From the result in Table 1., the model with BPE tokenization results in a better performance in Finnish but the performance of English and German dropped.

This shows that choosing the correct tokenization method can improve the generalization of the model for specific languages. However, since both the embedding layer and the overall model were still trained only on English data, the model struggled to effectively transfer knowledge to German and Finnish. The drop in performance for English and German suggests that while BPE tokenization better aligns with Finnish morphology, it may introduce suboptimal tokenization for English and German, leading to weaker representations for these languages.

Metric	Baseline RNN	Word-based Transformer	BPE-Trained Transformer
Multilingual			
F1-score	0.4992	0.8553	0.8136
Precision	0.5071	0.8630	0.8166
Recall	0.5078	0.8559	0.8138
Accuracy	0.5171	0.8561	0.8141
English (ENG)			
F1-score	0.4841	0.9144	0.8633
Precision	0.5010	0.9143	0.8636
Recall	0.5011	0.9145	0.8634
Accuracy	0.4906	0.9144	0.8633
Finnish (FIN)			
F1-score	0.2940	0.1407	0.3649
Precision	0.4829	0.5733	0.5106
Recall	0.4900	0.5065	0.5227
Accuracy	0.3050	0.1561	0.3993
German (GER)			
F1-score	0.4898	0.4555	0.5368
Precision	0.5011	0.5631	0.5572
Recall	0.5007	0.5172	0.5457
Accuracy	0.6840	0.6239	0.5854

Table 1: Performance comparison between Baseline RNN, Encoder-Only Transformer, and BPE-Trained Transformer on Multilingual Toxicity Detection development set.

2.2 Next Steps

To address the performance drop for English and German with BPE tokenization, we will focus on several key improvements. Firstly, we will try to experiment with language specific tokenization methods, such as a SentencePiece [14] tokenizer trained on a multilingual corpus. This is done so that we can better handle tokenization for Finnish and German.

To optimize even more for multilingual performance, we will adjust model complexity by experimenting with different architectures. This incudes modifications to attention blocks and hyperparameters such as learning rate, batch size, and weight decay. Early stopping will also be implemented to avoid overfitting.

To improve the model's performance in German and Finnish, we will incorporate additional training data in these languages. This could potentially be done by combining the training data and the dev data, then splitting that combination into a train/validation set split. Although it is not explicitly stated in the project/competition instructions that this is not allowed, we may test with this as it does provide crucial training data in Finnish and German.

As an additional step, we will implement a pretrained BERT [4] model and tokenizer, specifically testing the `BertForSequenceClassification` model. By having a model that has already been trained on multilingual data, we expect to improve accuracy in understanding and processing text across different languages. We have already fine-tuned a pretrained BERT model with one epoch and achieved great results as shown in Table 2. Next steps include fine-tuning it further, trying different methods, and trying to create a custom transformer solution that matches the pretrained BERT's scores.

Metric	**Fine-Tuned BERT Model**
Multilingual	
F1-score	0.8019
Precision	0.7479
Recall	0.8642
Accuracy	0.8286
English (ENG)	
F1-score	0.9401
Precision	0.9279
Recall	0.9527
Accuracy	0.9393
Finnish (FIN)	
F1-score	0.8599
Precision	0.8355
Recall	0.8858
Accuracy	0.7636
German (GER)	
F1-score	0.5503
Precision	0.4640
Recall	0.6761
Accuracy	0.7265

Table 2: Performance of the fine-tuned BERT model: BERTForSequenceClassification. With BERT tokenizer: BertTokenizer.

These efforts aim to enhance the overall performance of the model while maintaining strong Finnish/German language accuracy. As for improving the fine-tuned pretrained BERT model in Table 2, we can clearly see that the German scores are fairly low. Thus, we must

figure out ways to improve the German scores while maintaining similar rated scores for Finnish and English.

References

[1] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (K. Erk and N. A. Smith, eds.), (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, Aug. 2016.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2017.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[5] G. Lample and A. Conneau, "Cross-lingual language model pretraining," 2019.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] P. Contributors, "Transformerencoder pytorch," 2024.

[13] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016.

[14] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” 2018.

[15] European Language Resource Coordination (ELRC), “English-Finnish Corpus from Finnish Information Bank,” 2017. Last accessed: March 19, 2025.

[16] Wortschatz, University of Leipzig, “German Corpus from Wortschatz Leipzig,” 2024. Last accessed: March 19, 2025.