Distinguishing Between Cross-Domain and In-Domain Hate Speech Detection

Wafaa Aljbawi, Andreea Hazu, Hein Kolk, and Jelle Wassenaar

Vrije Universiteit, Amsterdam

1 Introduction

When looking at a twitter thread, it doesn't take much effort to find hateful messaging. People, often feeling protected behind their computer screens, are more likely to type hateful messages than they would have been able to say in real life. To tackle this growing trend, hate speech finding algorithms have been created. This paper studies the performance between these different algorithms on this specific task.

As the performance of such algorithms across different topical domains can vary quite heavily, this study aims to develop and execute different types of models for hate speech detection. These include transformer-based models, naive bayes, support vector machines and convolutional neural networks. To further gain insight into their performance irrespective of the data used, their performance is evaluated both for in-domain and cross-domain experimental setups. The in-domain experimental setup uses the OLID dataset [15] as training and test data and the cross-domain setup uses the HASOC [7] as train data and the OLID dataset as test data. Further, a quantitative error analysis for both setups is performed and the differences between these outcomes is discussed.

2 Dataset

2.1 OLID Dataset

The OLID dataset, also known as the Offensive Language Identification Dataset, has been made freely accessible online in 2019 [15]. It serves as a good example of how to annotate offensive content utilising a three-layer, fine-grained annotation approach that includes the detection of offensive language, categorization of offensive content, and identification of offensive language targets. Based on very specific API searches, Twitter was used to gather the dataset's instances. All tweets were in English. Each post or tweet is manually annotated by two, and occasionally by three annotators in case of disagreements. Furthermore, the dataset contains 13,240 tweets that have been annotated using the original level A scheme, which distinguishes between offensive and non-offensive tweets. However, this dataset has been pre-processed so that the label '1' refers to offensive messages and the label '0' corresponds to non-offensive messages. For this study, a subset of the OLID train dataset of the same size and label distribution as the HASOC train dataset was chosen. According to Table 1, the

test dataset contains 860 posts, with an unbalanced distribution of the classes (240 or 27.91% of offensive posts and non-offensive posts 620 or 72.09%), compared to 5852 posts in the OLID small training dataset (divided into 2261 or 38.64% offensive posts and 3591 or 61.36% of non-offensive posts). Besides this, in the non-offensive and offensive classes of the OLID small train set, the average message length was 123.09 and 131.37 characters, respectively. In the OLID test set, the average message length was longer for both classes. Nonetheless, the average number of words per message and average word length for both classes in the train and test sets were almost the same.

	OLID-train-small		OLID-test		HASOC-train	
Class	0	1	0	1	0	1
Number of instances	3591	2261	620	240	3591	2261
Relative label frequency	61.36%	38.64%	72.09%	27.91%	61.36%	38.64%
Average message length	123.09	131.37	148.90	138.90	164.42	171.22
Average number of words per message	12.087	11.731	12.861	11.379	12.319	11.905
Average word length	4.377	4.371	4.834	4.804	6.038	5.841

Table 1. Statistics for distribution of classes in the 3 datasets

2.2 HASOC Dataset

Hate Speech and Offensive Content Identification (HASOC) in Indo-European Languages, specifically German, English, and Hindi. Twitter and Facebook were used to generate the datasets for all three languages [7]. The goal of HASOC is to inspire research and development for Hate Speech classification in several languages, as well as to facilitate the creation and testing of supervised machine learning algorithms. HASOC is divided into three tasks: a coarse-grained binary classification task (classification of hate speech (HOF) and non-offensive content) and two fine-grained multi-class classifications (if the post is HOF, identify the type of hate and the target of the post). Only the training set of the English dataset, which consisted of tweets and Facebook messages, was used in this work. The dataset was pre-processed similar to the OLID dataset, with label '1' corresponding to hateful/offensive messages and label '0' referring to non-hateful/non-offensive messages. As mentioned earlier, the classes distribution of OLID small training and HASOC is the same, with 61.36% non-offensive and 38.64% offensive messages. According to Table 1, the average message length was longer than the OLID train and test sets, with 164.42 and 171.22 characters in the non-offensive and offensive classes, respectively. Correspondingly, the average number of words per message is similar to the other datasets. Finally, the average word length was somewhat greater, with 6.038 and 5.841 for the non-offensive and offensive classes, respectively.

3 Methods

The used methods are thoroughly described in this section. Along with a thorough and detailed discussion of the modeling choices.

3.1 Transformer-based Models

A variety of research has been conducted to study the use of machine learning algorithms, such as RNN, for hate speech detection. However, these algorithms suffer from intrinsic problems such as long-term dependency and a lack of parallelization [11]. To tackle these issues, transformer-based models were developed [13]. These models perform sequence transduction entirely relying on attention. Such models enable the capture of important information that may be present in each word of a sentence, while simultaneously also enabling parallel processing. Take the following example as an illustration: *"Foreigners must go. They're taking over our jobs. Some of them are stealing us"* [11]. Current techniques, which are primarily based on basic deep learning algorithms, fail to detect that the term *"Some"* in the third sentence refers to foreigners [11]. This is because they rely on past hidden states to identify dependencies with preceding words [11]. Whereas, transformer models use positional embedding to remember word order in sequences to capture such long-term relationships.

In NLP research, numerous types of transformer approaches have been successfully examined. These methods include BERT, RoBERTa, DistilBERT and XLNET. BERT creates models whose parameters may be modified to enhance performance on smaller amounts of supervised data [2]. RoBERTa is an improved version of BERT that is trained on a larger dataset to enhance performance [6], whereas XLNET is a generalised autoregressive pre-training approach that attempts to rebuild original data from corrupted input [14].

This paper uses BERT, RoBERTa, and XLNET to assess the efficacy of a Transformer-based model on hate speech detection.

3.2 Other Models

Naive Bayes

The Naive Bayes (NB henceforth) algorithm is based on Bayes' Theorem known from Probability Theory. Bayes' Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. In machine learning, NB is typically used for classification problems. The NB approach is popular due to its simplicity, its linear computational complexity and its accuracy ([10]). The Pandey et al. [12] paper implemented NB, Support Vector Machines (SVM, section 3.2) and other Machine Learning approaches. The authors concluded that only SVM was able to produce better results than NB. In this paper, the validity of this conclusion will be tested.

Modeling Choices: The multinomial model of Naive Bayes is chosen, as this provides the ability to perform the classification using only relatively small datasets. The TF-IDF (Term Frequency - Inverse Document Frequency) is used as the weighting scheme for the multinomial Naive Bayes model. There were no other modeling choices made.

Support Vector Machines

This algorithm creates a line or a hyperplane which separates the data into classes. Support vector machines (SVM henceforth) can solve classification as well as regression problems. The data is transformed with the use of a linear

function and the use of an activation function. Each input message is defined by a TF-IDF (Term Frequency Inverted Document Frequency) vector. As stated in the previous section, SVM was able to outperform NB in some tasks. However, Caselli et al. [1] shows that the SVM approach was also outperformed by a Transformer-based model, namely the BERT model.

Modeling Choices: The linear model of the SVM algorithm was chosen, as it is less prone to overfitting than non-linear. As NB, the SVM algorithm also made use of the TF-IDF weighting scheme. Moreover, L1 regularization was used for SVM.

Convolutional Neural Networks

A Convolutional Neural Network (CNN henceforth) is a Machine Learning approach which can be used for different applications and data types. The deep learning CNN consists of three different layers:

- A Convolutional Layer, which involves a kernel or filter which checks if a feature is present.
- A Pooling Layer, which reduces the number of parameters present in the model.
- A Fully Connected Layer, where all the inputs or nodes from one layer are connected to the activation function of the next layer.

Each layer of the CNN learns different information about the data. At each successive layer, the complexity increases and the algorithm is able to identify more difficult features that uniquely represent the input. Several papers have already done research in the ability of CNN to predict hate speech. Gamback et al. [4] concluded that CNN was able to outperform a Logistic Regression model. Also, Markov et al. research [8] produced an experiment in the same manner as this paper proposes; also with an in-domain and cross-domain setting. The results in the paper showed that CNN performed almost the same as SVM, however both algorithms were outperformed by Transformed-Based models and Ensemble methods.

Modeling Choices:For the CNN algorithm instead of a TF-IDF vectorizer a Tokenizer was chosen. ADAM optimization was chosen. With trial and error, 20 epochs were chosen along with a batch size of 10.

4 Experimental Setup

In order to gain further insight into the performance irrespective of the data used, the performance of the models is evaluated both for in-domain and cross-domain experimental setups. The next two sections further explain these two setups.

4.1 In-domain Experiments

In the in-domain setting, a classifier is trained and evaluated on data from the same domain. In general, we anticipate the models employed to perform reasonably well on this setting, since the model will be trained and tested on a

very similar dataset. To get a better insight into how the hate speech detecting models that will be evaluated perform in such scenarios, we trained all models described in Section 3 on a subset of the OLID dataset and evaluate on a different subset of the same dataset.

4.2 Cross-domain Experiments

In the cross-domain setting, a classifier is trained on data from one domain and tested on data from another. According to [5] the accuracy of machine learning algorithms is frequently poorer in cross-domain settings due to the distribution gap between domains. Consequently, in order to determine how such settings would impact the performance of all of our models, they were trained on the HASOC train dataset and tested on a subset of the OLID dataset.

5 Results and Analysis

For each experiment, the precision, recall, F1-score, macro-averaged scores and confusion matrices obtained are reported in this section together with a thorough quantitative analysis of the models' performances. All confusion matrices can be found in the appendix.

5.1 Transformer-based Models

When evaluated on an in-domain setting, BERT successfully identified 552 out of 620 non-offensive messages and 162 out of 240 offensive messages, as shown in Figure 1. RoBERTa on the other hand, detects fewer offensive messages than BERT, with a total of 145 out of 240 offensive messages and 580 out of 620 non-offensive messages, as illustrated in Figure 2. Lastly, Figure 3 shows that XLNet predicts 553 out of 620 of the non-offensive messages and 160 out of 240 of the offensive ones. Additionally, this is evident in the outcomes in Table 2. We can observe that RoBERTA had the best precision on the offensive class among the other models, scoring 0.78, but BERT had a greater recall on the same class, scoring 0.68. The highest F1-score on the offensive class was 0.69 for both for BERT and XLNet.

For the other class, BERT gave the highest precision with a score of 0.88. However, the highest recall and F1 scores were achieved by RoBERTa. Macro-averaged F1-score was utilised to determine the best performing model. In conclusion, the in-domain experiment results showed that both BERT and RoBERTA performed remarkably well on the dataset and can properly recognise 561 non-offensive messages and 137 offending messages. Surprisingly, RoBERTa was unable to detect any offensive messages, but accurately identified all non-offensive messages, as shown in Figure 2. On the contrary, we can see from Figure 3, XLNet was quite successful at finding non-offensive messages but only found 85 of the offending ones.

Different pre-trained models are tested in a large cross-domain experiment. As demonstrated in Figure 1, BERT can
and take contextual information into consideration.

Accordingly, Table 3 also reported similar results. BERT had the greatest precision, recall, and F1 scores for the offensive class, with scores of 0.70, 0.57,

	OFF			NON			Macro-averaged		
Model	P	R	F1	P	R	F1	P	R	F1
BERT	0.70	0.68	0.69	0.88	0.90	0.88	0.79	0.78	**0.79**
RoBERTa	0.78	0.60	0.68	0.86	0.94	0.90	0.82	0.77	**0.79**
XLNet	0.70	0.67	0.69	0.87	0.89	0.88	0.79	0.78	0.78

Table 2. The performance of transformer-based models for in-domain settings in terms of precision, recall, and F1-score for each class and macro-averaged results

and 0.63, respectively. As expected, RoBERTa achieved a 0 on all class metrics. BERT again had the highest precision and F1 scores in the non-offensive class, while RoBERTa had the highest recall. BERT remains the best-performing model in the cross-domain and in-domain setting, with a macro-averaged score of 0.75. RoBERTa, on the other hand, exhibits a considerable decline in macro averaged F1-score from 0.79 (in-domain settings) to 0.42 (corss-domain settings), which may be attributed to the relative complexity of the HASOC hateful content, particularly for identifying offensive messages. A similar effect was discovered by [3], where similar behaviour was detected after training on toxic messages using Ask.fm as an out-of-domain test set [9].

The cross-domain findings show that using out-of-domain data resulted in a significant drop in performance ranging from 4% to 37% F1 points for all models tested. This was also aligned with the findings of [8]. The different topical focuses of these datasets may be one of the explanations for this decline in performance in the in-domain and cross-domain scenarios. In addition, the length of messages varied significantly throughout the HASOC train, OLID train, and test set. According to [9], there is a noticeable tendency for the ensemble approach to successfully identify the longer instances, whereas BERT performs better on shorter instances [9]. This explains why BERT, RoBERTA, and XLNET performed poorly during the cross-domain experiment because the messages in the HASOC train were longer than the other datasets.

	OFF			NON			Macro-averaged		
Model	P	R	F1	P	R	F1	P	R	F1
BERT	0.70	0.57	0.63	0.85	0.90	0.87	0.77	0.74	**0.75**
RoBERTa	0.0	0.0	0.0	0.72	1.0	0.84	0.36	0.50	0.42
XLNet	0.56	0.35	0.43	0.78	0.89	0.83	0.67	0.62	0.63

Table 3. The performance of transformer-based models for cross-domain settings in terms of precision, recall, and F1-score for each class and macro-averaged results

5.2 Other models

Naive Bayes

When evaluated on an in-domain experimental setup Naive Bayes performed quite well, gaining a precision of 0.92 and 0.76 on offensive and non-offensive classes, respectively (for in-domain setting, see Table 4). As can be seen in Appendix 4, it correctly classified 616 out of 620 non-offensive tweets as be-

ing non-offensive. Furthermore, it classified 45 out of 140 tweets correctly as being offensive. When using a cross-domain experimental setup, Naive Bayes performed similarly. It obtained a precision score of 0.75 by correctly classifying 609 out of 620 non-offensive tweets and score of 0.77 for detecting 36 out of 240 offensive tweets (for cross-domain setting, see Table 5).

Analysis

It can easily be seen that Naive Bayes performs quite well because it classifies nearly everything as non-offensive. This improves the accuracy score quite heavily because the test set consists of largely non-offensive tweets. Mostly, the actual label being offensive and a prediction of non-offensive are classified wrong. Again, looking at the Appendix 4, for the in-domain setting this happened 195 times, whereas the prediction of offensive and the classification of non-offensive only occurred 4 times. For the cross-domain setting, the actual label being offensive and the prediction of non-offensive occurred 204 times and the prediction of offensive and the classification of non-offensive only occurred 11 times. Therefore, the recall for the offensive tweets for both the in-domain and cross-domain setting was quite low, 0.19 & 0.15 respectively. For the non-offensive tweets, the recall becomes very high, namely 0.99 & 0.98. In the in-domain setting, the Naive Bayes model obtained a far better score than the cross-domain in terms of accuracy (0.92 & 0.77) which resulted in a higher macro F1 score for the in-domain setting (0.59 vs. 0.55). This can be explained due to the fact that the Naive Bayes algorithm is not able to overcome the distribution gap created by the cross-domain setting.

	OFF			NON			Macro-averaged		
Model	P	R	F1	P	R	F1	P	R	F1
Naive Bayes	0.92	0.19	0.31	0.76	0.99	0.86	0.84	0.59	0.59
SVM	0.71	0.47	0.57	0.82	0.93	0.87	0.76	0.70	**0.72**
CNN	0.49	0.36	0.42	0.78	0.85	0.81	0.63	0.61	0.61

Table 4. The performance of the other models for in-domain settings in terms of precision, recall, and F1-score for each class and macro-averaged results

Support Vector Machines

Taking a look at Appendix 5, the SVM model managed to correctly classify 574 out of 620 non-offensive tweets when evaluated on an in-domain experimental setup, which provided an accuracy score of 0.80 (for in-domain setting, see Table 4). Furthermore, it correctly classified 113 out of 240 offensive tweets. When using a cross-domain experimental setup, the algorithm classified 435 out of 620 tweets correctly as being non-offensive, obtaining an accuracy score of 0.64 (for cross-domain setting, see Table 5). Furthermore, it correctly classified 117 out of 240 tweets as being offensive.

Analysis

The Support Vector Machine algorithm was able to perform better than the Naive Bayes and the CNN algorithms. From Appendix 5, it can be deducted that the SVM algorithm misclassifed in the in-domain setting 127 labels as Non-Offensive and 46 labels as Offensive. For the cross-domain setting 123 labels were misclassified as Non-Offensive and 185 labels as Offensive. Therefore, also

for SVM the in-domain setting yielded better results in terms of the Macro F1 score than the cross-domain setting. Again, as seen in the Naive Bayes model, mostly tweets are predicted as Non-Offensive. However, much fewer than for the Naive Bayes algorithm. This resulted in a lower precision for offensive tweets, 0.71 & 0.39 for in-domain and cross-domain respectively, but a much higher recall: 0.47 & 0.49 for in-domain and cross-domain respectively. The accuracy for the non-offensive tweets for the in-domain were calculated at 0.82 which was higher than the 0.78 for the cross-domain setting. However, the recall of the cross-domain setting was quite low: 0.70. The recall of the in-domain setting was higher and calculated at 0.93. The explanation for the lower recall for the non-offensive tweets and a higher recall for the offensive tweets for both the in-domain as the cross-domain setting in comparison with the Naive Bayes model can be found in the fact that the SVM model is better at detecting the 'positive' samples of the offensive tweets than the Naive Bayes model. When looking at the F1 macro score the SVM model outperformed the NB model, which was in line with the results found by the Pandey et al. paper [12]. Furthermore, the SVM model in the cross-domain setting wrongly predicted a higher amount of non-offensive tweets than offensive than it did in the in-domain setting (185 vs. 46 respectively). The SVM in-domain experiment yielded better results than the SVM cross-domain, which again could be due to the fact that the SVM model is also unable to overcome the distribution gap created by the cross-domain setting.

	OFF			NON			Macro-averaged		
Model	P	R	F1	P	R	F1	P	R	F1
Naive Bayes	0.77	0.15	0.25	0.75	0.98	0.85	0.76	0.57	0.55
SVM	0.39	0.49	0.43	0.78	0.70	0.74	0.58	0.59	**0.59**
CNN	0.32	0.37	0.34	0.74	0.70	0.72	0.53	0.53	0.53

Table 5. The performance of the other models for cross-domain settings in terms of precision, recall, and F1-score for each class and macro-averaged results

Convolutional Neural Networks

According to Appendix 6, the CNN managed to correctly classify 528 out of 620 non-offensive tweets when evaluated on an in-domain experimental setup, which provided an accuracy score of 0.72 (for all in-domain metrics, see Table 4). Furthermore, it correctly classified 87 out of 240 offensive tweets. When using a cross-domain experimental setup, the algorithm classified 432 out of 620 non-offensive tweets and 89 out of 240 offensive tweets. Furthermore, it gained an accuracy score of 0.61 (for all cross-domain metrics, see Table 5)

Analysis

The CNN algorithm performed similarly to the Naive Bayes algorithm, but was outperformed by the SVM algorithm. Furthermore, CNN predicted most labels as Non-Offensive. However, in the cross-domain setting a higher amount of non-offensive tweets were incorrectly predicted as offensive than in the in-

domain setting (188 vs. 92 respectively). The CNN algorithm was able to obtain a higher recall for the offensive tweets than the Naive Bayes algorithm, however the SVM algorithm still obtained the highest recall for offensive tweets. The CNN algorithm performed poorly on the precision of the offensive tweets, 0.49 & 0.32 for the in-domain and cross-domain setting respectively. Also, the recall for the offensive tweets was not high (0.36 & 0.37 for in-domain and cross-domain setting). For the non-offensive tweets, the CNN in-domain setting yielded higher results than for the cross-domain setting. In comparison with the metrics for the non-offensive predictions of the other models, for the in-domain setting the CNN algorithm performed decently. Also for the cross-domain setting, the CNN algorithm performed decently but all other algorithms obtained better results. This could be due to the fact that the CNN algorithm was not able to detect non-offensive tweets. Furthermore, as for the Naive Bayes and the SVM algorithm a possible explanation for the the CNN algorithm worse performance in the cross-domain compared to the in-domain setting, a F1 Macro score of 0.59 & 0.72 respectively, is that the CNN algorithm is not able to overcome the distribution gap created by the cross-domain setting. Compared with the SVM algorithm, the CNN algorithm was not able to perform as good or better as the SVM model in terms of F1 Macro score. This was not in line with the findings from the Gambäck et al. paper [4]. A possible explanation for this could be a different choice of hyperparameters for the CNN algorithm. Future research is needed to address this problem.

5.3 Comparison Transformer vs Non Transformer

Looking at Table 2 & Table 4 and comparing the in-domain setting for the transformer and the non-transformer models, the conclusion can be drawn that the transformer models perform much better than the non-transformer models. The reason for this is the ability of transformer models to use positional embedding to remember word order in sequences to capture long-term relationships. Taking a look at the cross-domain setting, Table 3 and Table 5 show that because of the RoBERTa model the transformers and non-Transformer models perform almost even, as the RoBERTa model performed very poorly on the classification of offensive tweets. However, it can be noted that the BERT and XLNet model were able to outperform all non-transformer models. Again, the same reasoning that transformer models use positional embedding to remember word order in sequences to capture long-term relationships can be used.

6 Conclusion

In order to develop an effective model for hate speech detection, machine learning models can be used to automatically tackle this problem. The goal of this study was to examine several approaches for enhancing these detection algorithms, and numerous experiments were carried out to assess the effectiveness of such models in both in-domain and cross-domain experimental setups. Based on our results, transformer models outperformed other basic models such as SVM

or NB. We also find that BERT was the best model for both the in-domain and cross-domain setups. Nonetheless, more research into alternative approaches such as ensemble models for hate speech detection might be conducted in order to improve on the findings of this study.

References

[1] Tommaso Caselli et al. "DALC: the Dutch Abusive Language Corpus". In: *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*. 2021, pp. 54–66.

[2] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[3] Chris Emmery et al. "Current limitations in cyberbullying detection: On evaluation criteria, reproducibility, and data scarcity". In: *Language Resources and Evaluation* 55.3 (2021), pp. 597–633.

[4] Björn Gambäck and Utpal Kumar Sikdar. "Using convolutional neural networks to classify hate-speech". In: *Proceedings of the first workshop on abusive language online*. 2017, pp. 85–90.

[5] Radu Tudor Ionescu and Andrei Madalin Butnaru. "Transductive learning with string kernels for cross-domain text classification". In: *International Conference on Neural Information Processing*. Springer. 2018, pp. 484–496.

[6] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[7] Thomas Mandl et al. "Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages". In: *Proceedings of the 11th forum for information retrieval evaluation*. 2019, pp. 14–17.

[8] Ilia Markov and Walter Daelemans. "Improving cross-domain hate speech detection by reducing the false positive rate". In: *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*. 2021, pp. 17–22.

[9] Ilia Markov et al. "Exploring stylometric and emotion-based features for multilingual cross-domain hate speech detection". In: *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 2021, pp. 149–159.

[10] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. "Spam filtering with naive bayes-which naive bayes?" In: *CEAS*. Vol. 17. Mountain View, CA. 2006, pp. 28–69.

[11] Raymond T Mutanga, Nalindren Naicker, and Oludayo O Olugbara. "Hate speech detection in twitter using transformer methods". In: *International Journal of Advanced Computer Science and Applications* 11.9 (2020).

[12] Yogesh Pandey et al. "Hate Speech Detection Model Using Bag of Words and Naıve Bayes". In: *Advances in Data and Information Sciences*. Springer, 2022, pp. 457–470.

[13] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).
[14] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019).
[15] Marcos Zampieri et al. "Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval)". In: *arXiv preprint arXiv:1903.08983* (2019).

7 Appendix

7.1 Appendix A: Who Did What

Name	Tasks
Wafaa Aljbawi	Report (Datasets, Transformed-based models analysis and description, Experimental setup, Conclusion) and Code (Class distribution and analysis, Implementation and experiments of transformer models)
Andreea Hazu	Report (Datasets, Transformed-based models analysis and description, Experimental setup, Conclusion) and Code (Class distribution and analysis, Implementation and experiments of transformer models)
Hein Klok	Report (Non-Transformer Models, both set up and analysis, Comparison Transformer vs Non Transformer and Introduction) and Code (Implementation Non-Transformer Models)
Jelle Wassenaar	Report (Non-Transformer Models, both set up and analysis, Comparison Transformer vs Non Transformer) and Code (Implementation Non-Transformer Models)

Table 6. list of who did what in this assignment

7.2 Appendix B: BERT

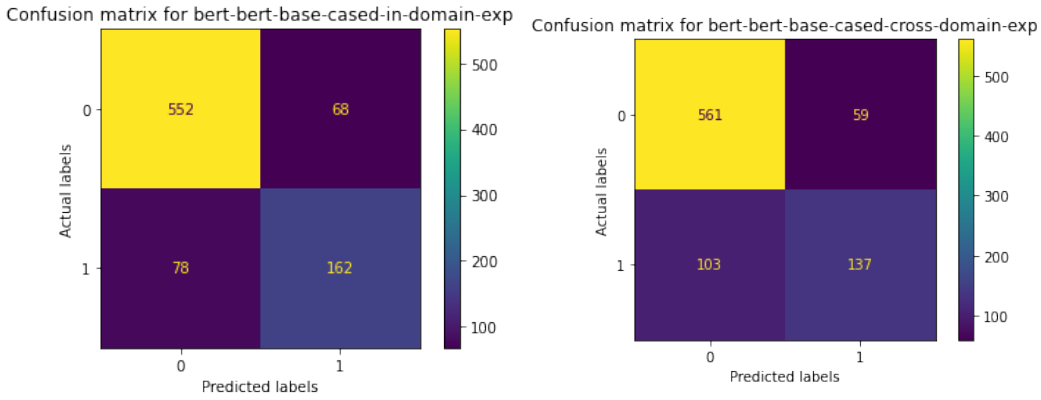

Fig. 1. Confusion matrix using BERT on in-domain (Left) and cross-domain (Right) datasets

7.3 Appendix C: RoBERTa

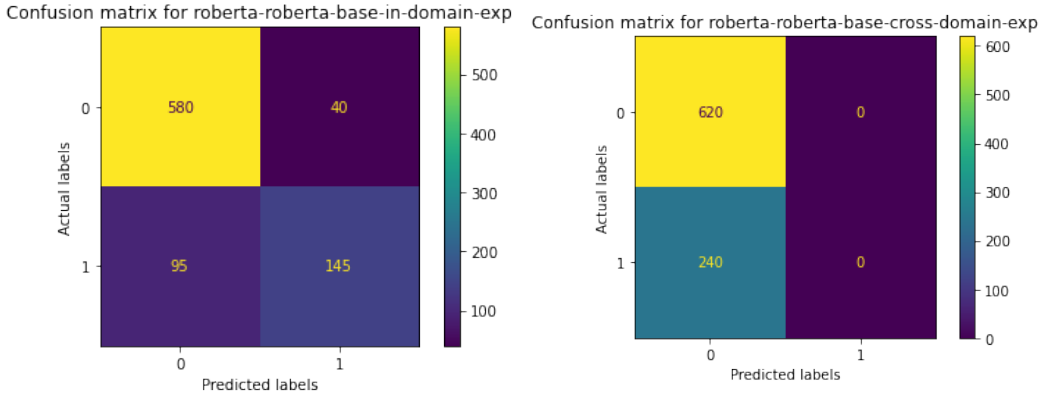

Fig. 2. Confusion matrix using RoBERTa on in-domain (Left) and cross-domain (Right) datasets

7.4 Appendix D: XLNet

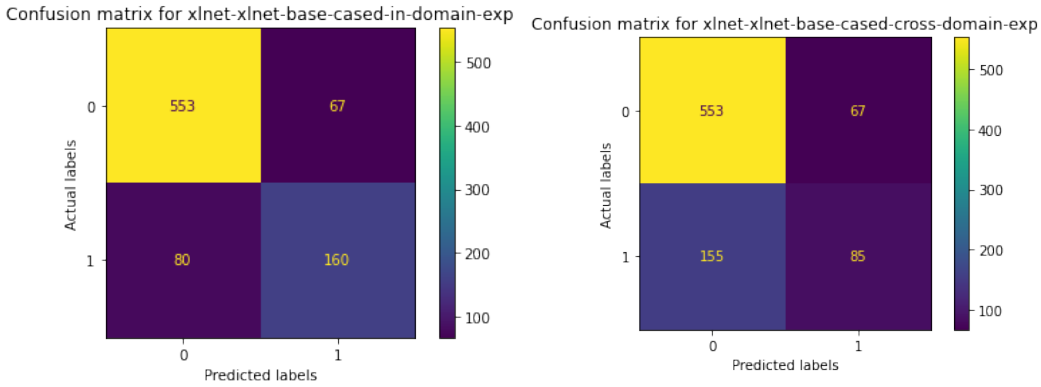

Fig. 3. Confusion matrix using XLNet on in-domain (Left) and cross-domain (Right) datasets

7.5 Appendix E: Naive Bayes

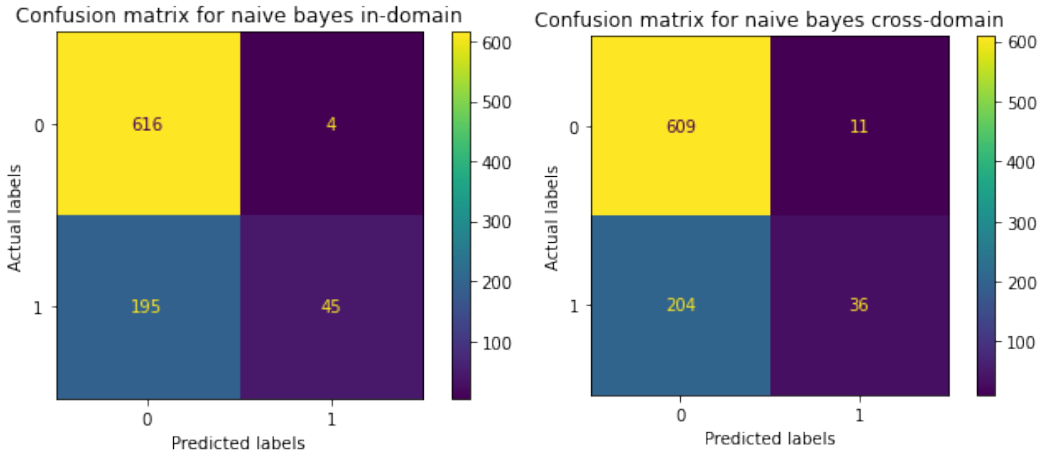

Fig. 4. Confusion matrix using Naive Bayes on in-domain (Left) and cross-domain (Right) datasets

7.6 Appendix F: Support Vector Machines

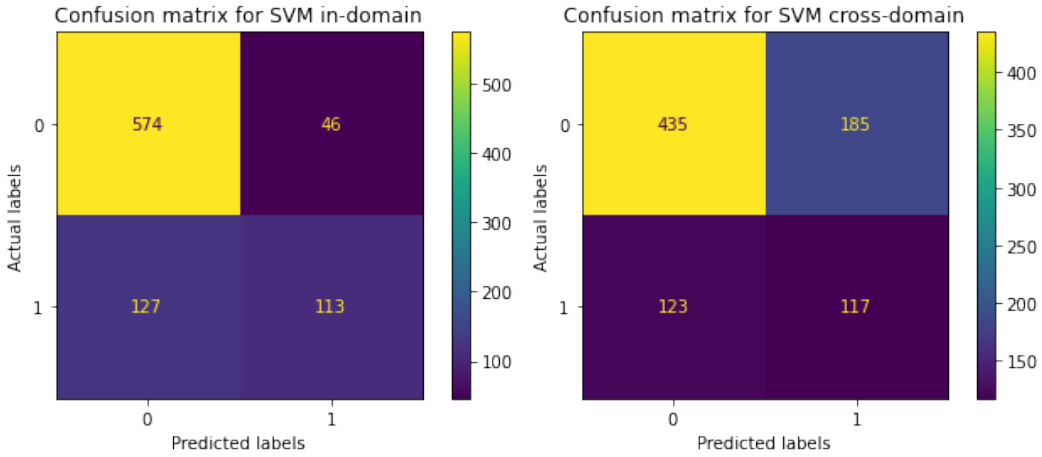

Fig. 5. Confusion matrix using Support vector machines on in-domain (Left) and cross-domain (Right) datasets

7.7 Appendix G: Convolutional Neural Networks

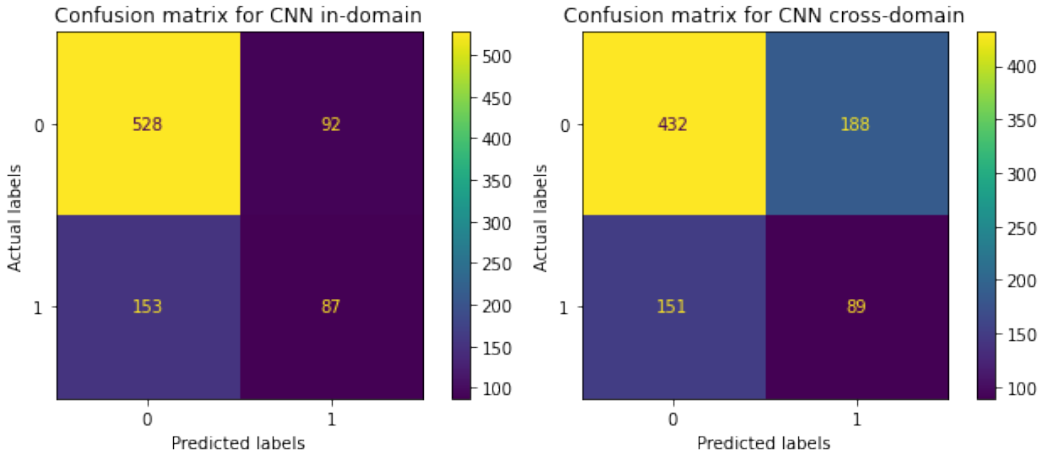

Fig. 6. Confusion matrix using Convolutional Neural Networks on in-domain (Left) and cross-domain (Right) datasets