Exploring Cross-Domain Effects in Targeted Sentiment Analysis for Norwegian

Anonymous ACL submission

Abstract

Targeted Sentiment Analysis (TSA) is a task of identifying a target of an opinion and classifying its corresponding polarity in a given text. This paper analysed the effects of training neural models for TSA while utilizing data for training and testing from (partly) heterogeneous domains or categories. The results demonstrated that models trained on domains containing universal content show adequate performance when tested on other domains, while some categories are inherently difficult. Additionally, the results illustrated that combining data from various domains yields enhanced performance.

1 Introduction

Sentiment Analysis (SA), also referred to as Opinion Mining is a Natural Language Processing (NLP) task, where the goal is to determine an opinion behind a text unit and classify its sentiment (e.g. positive or negative). Research in SA comprises various domains, such as analyzing movie reviews (Pang et al., 2002). While SA would define an overall sentiment of a review, TSA would provide entity-level sentiment for a specific entity. Given an example movie review "*Effects were great, but plot horrendous"*, the entities would be *effects* and *plot*, each assigned positive and negative sentiments, respectively. Taking into consideration the fact that TSA is a more fine-grained task, it is more precise, yet arguably more challenging than SA. Moreover, one of the major limitations of the usefulness of SA models, in general, is domain barriers. It is plausible to assume that a model trained on data originating from one domain (e.g. movie reviews) would demonstrate inferior performance when predicting sentiment in a different domain (e.g. sports).

This paper aimed to investigate the cross-domain effects in the task of TSA for Norwegian. More specifically, it focused on NoReC$_{\text{fine}}$ dataset (Øvrelid et al., 2020), exploring what domains appeared to be the most challenging to classify, how various combinations of data from heterogeneous domains affected the model's performance, as well as analyzed errors from particular domains.

The rest of the paper is structured as follows: Section 2 presents the data used for experiments. A description of the model selected for testing is given in Section 3. Section 4 surveys the experiments conducted. Section 5 summarizes and sets forth the findings and results of experiments. Error analysis is given in Section 6. Section 7 concludes the findings, before the paper is finished with Section 8, which concerns suggestions for future work.

2 Data

This paper made use of NoReC$_{\text{fine}}$, a dataset for fine-grained sentiment analysis in Norwegian. It comprises a subset of the Norwegian Review Corpus (Velldal et al., 2018). The text units in the dataset are annotated with not only binary polarity labels (positive/negative) but also intensity labels (slight, standard, strong), making the dataset fitting for both binary and more fine-grained, 6-way classification. In addition, it includes sentiment holders, polarity expressions, targets, and relationships between them. The underlying texts originate from professionally authored reviews from multiple news sources, including a variety of domains: 'literature', 'screen', 'sports', 'music', 'games', 'products', 'stage', 'restaurants', and 'miscellaneous'. Figure 1 illustrates the spread of data between domains.

For the purposes of this paper, the data came in conll-format, and included a total of 5 labels, represented as BIO-labels with additional polarity: B-targ-Positive (B-Pos), I-targ-Positive (I-Pos), B-targ-Negative (B-Neg), I-targ-Negative (I-Neg), and O. Table 1 demonstrates an example word sequence with its respective labels extracted from games domain. The data can be obtained in the

Token	Label
Slåssespill	B-Neg
er	O
i	O
sin	O
natur	O
repeterende	O

Table 1: Example sentence from NoReC$_{\text{fine}}$ 'games' domain. *Slåssespill* is an entity, labeled with negative sentiment. English translation of the sentence is as follows: '*Fighting games are, by their nature, repetitive*'.

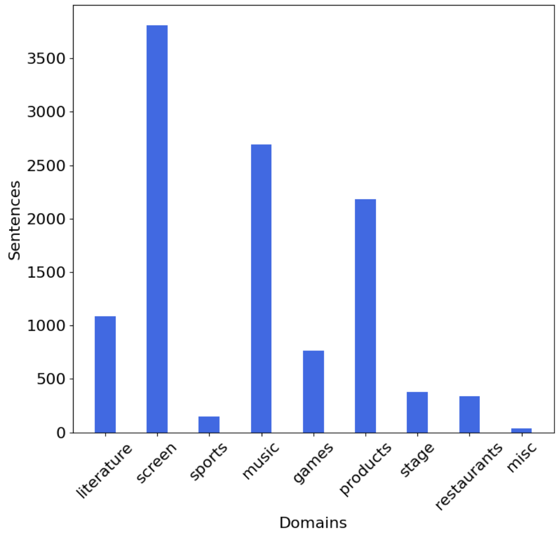

Figure 1: Spread of data in NoReC$_{\text{fine}}$. 'Screen' is most represented, while 'sports' and 'misc' comprise the fewest number of samples.

GitHub repository of the IN5550 course[1].

3 Models

For the task of TSA this paper employed BERT (Bidirectional Encoder Representations from Transformers) pre-trained language model (LM), introduced by (Devlin et al., 2019). More specifically, it made use of NorBERT$_3$ (Samuel et al., 2023b): large pre-trained contextualized LM for Norwegian, based on BERT architecture. The pre-training phase of NorBERT$_3$ encompassed various text collections: Norwegian Wikipedia dumps both for Bokmål and Nynorsk from October 2022, Public domain texts released by the National Library of Norway in 2015[2], Norwegian News Corpus[3], Norwegian Colossal Corpus[4], as well as Norwegian part of of web-crawled mC4 corpus (Xue et al., 2021). Moreover, the authors employed the masked language modeling approach for pre-training NorBERT$_3$ and followed the optimized training method from (Samuel et al., 2023a). This approach differs from the standard BERT training.

In the same manner, as BERT, NorBERT$_3$ comes in several versions, including models with various numbers of parameters. This paper conducted an experiment where several variants of NorBERT$_3$ were tested on the NoReC$_{\text{fine}}$ dataset mentioned in Section 2. A thorough description of the experiment and its results are provided in Section 4 and Section 5, respectively. The results of the experiment motivated the use of the NorBERT$_{3,\text{base}}$ model, which is the main model utilized in the following experiments. The model comprised 123M parameters, 12 hidden layers, 12 attention heads, and hidden dimensions of size 786.

4 Experimentation

Several experiments were carried out in this paper, where each one was conducted using the same model hyperparameters: batches of size **16**, **8** training epochs, learning rate of **3e-5**, fine-tuning all of the models' parameters.

The first experiment described in subsection 4.1 concerned the selection of the NorBERT$_3$ model for further analysis. Subsection 3.2 surveys how using different domains, as well as their combination affected model performance. An investigation of the impact of adding a fixed number of samples from target domains to training data is presented in subsection 3.3. The final experiment with adding different ratios of data from different domains is described in subsection 3.4.

4.1 Model selection

As mentioned in Section 3, several versions of NorBERT$_3$ were extant at the moment of writing of this paper. The goal of the experiment was to determine the best-performing model to use in further analysis. In particular, 4 different versions of the model were tested: NorBERT$_{3,\text{small}}$, NorBERT$_{3,\text{base}}$, NorBERT$_{3,\text{large}}$,

[1] https://github.uio.no/in5550/2023/tree/main/exam/tsa

[2] https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-34/

[3] https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-4/

[4] https://huggingface.co/datasets/NbAiLab/NCC

and NorBERT$_{3,\text{wiki-base}}$. A baseline was established for each model by retaining 80% of the data from the 'screen' domain for training while using the resting 20% for the purposes of testing. This made it feasible to carry out a comparison between the way models behaved when trained and tested on data originating from identical or dissimilar domains. The rationale behind selecting the 'screen' domain as training data was due to the domain being the most prevalent one. Consequently, the most efficient and best-performing model was picked for further experiments.

4.2 Merging and training on different domains

The second experiment came down to testing whether using a different domain as training data would potentially prove useful. The 'products' domain was of particular interest. As described by the authors of the NoReC$_{\text{fine}}$ dataset, the most diverse category is perhaps the 'products' category, which comprises product reviews across a number of subcategories. Establishing an identical baseline of training the model on 80% of the 'product' domain data and testing on the remaining 20%, the model was then further tested on all of the remaining domains.

Furthermore, it seemed intuitively plausible that combining data from different domains and using that data for training could potentially increase the model's performance. Consequently, another experiment was conducted where the 'screen' and 'product' domains were merged together and used for training the model, while data from the remaining domains was utilized for testing purposes. Whereas the screen domain contained the most amount of data, the product domain served as the most "universal" category.

4.3 Adding a fixed number of data from target domain to training data

In this experiment, the aim was to investigate the effect of adding a fixed amount of samples from the target domains to the training data. In particular, two tests were conducted: adding 10 and 100 samples. Furthermore, it was determined to drop the misc category in the testing phase, as it contained way too few samples for reliable performance assessment.

Model	**F1-score**
NorBERT$_{3,\text{small}}$	0.4836
NorBERT$_{3,\text{base}}$	**0.4842**
NorBERT$_{3,\text{large}}$	0.4807
NorBERT$_{3,\text{wiki-base}}$	0.4018

Table 2: Baseline performance of 4 different NorBERT$_3$ models trained on 80% and tested on 20% of the data from 'screen' domain.

4.4 Adding ratio of data from all domain to training data

This experiment focused on testing whether adding a certain ratio of data from each category (depending on the total number of samples in that category) to training data would prove useful. More specifically, 3 different ratios were considered: taking **0.1**, **0.2**, and **0.3** of total data in each domain and adding it to the training data, while utilizing the rest for testing. Additionally, 3 different random seeds were employed for each run, ensuring some randomness when sampling data for training, and resulting in 9 runs in total. Consequently, the experiment results ended up being more reliable, owing to the fact that the testing subset was slightly different between each run.

All the results for the aforementioned experiments are presented in Section 5.

5 Results

This section gives a report and surveys the results of experiments described in Section 4.

5.1 Experimenting with different NorBERT$_3$ models

Results for baseline testing can be found in Table 2. While NorBERT$_{3,\text{base}}$ yielded the highest Batista F1-score[5], both NorBERT$_{3,\text{small}}$ and NorBERT$_{3,\text{large}}$ underperformed only slightly.

As illustrated in Figure 2, NorBERT$_{3,\text{base}}$ and NorBERT$_{3,\text{large}}$ demonstrated highest F1-scores during tests in other domains. While the latter model tended to yield better performance in domains like 'games', 'restaurants', and 'stage', NorBERT$_{3,\text{base}}$ outperfromed it in the rest of the domains.

Given the fact that NorBERT$_{3,\text{base}}$ contained sufficiently fewer parameters than its expanded

[5] https://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/

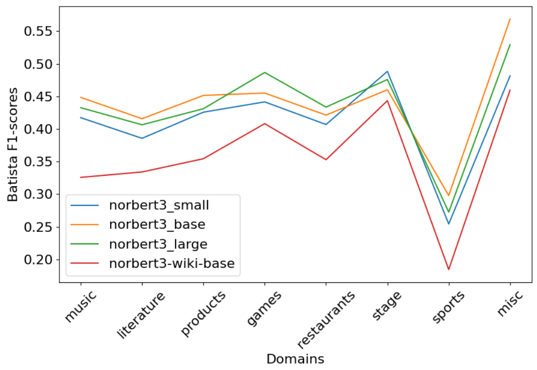

Figure 2: Batista-F1 scores achieved by each model trained on 'screen' domain. NorBERT$_{3,\text{base}}$ and NorBERT$_{3,\text{large}}$ appear to be the most performant models.

version NorBERT$_{3,\text{large}}$ (123M vs. 353M) while yielding similar performance, the choice of the model for further experiments ended on the first one. Moreover, the NoReC$_{\text{fine}}$ is a dataset of arguably moderate size, which motivated the use of a less parameterized model to avoid any potential side effects of overfitting.

5.2 Investigating the use of various single and merged domains as training data

The results for testing the effect of using 'products' domain as training data, as well as merging together 'screen' and 'product' domains are illustrated in Table 3. Despite the fact that 'products' domain is perhaps the most diverse one, the model fine-tuned on it didn't manage to perform significantly better than the same model fine-tuned on data from 'screen' domain. While the model showed better performance in 'games' domain (a **0.0284** increase) and 'restaurants' domain (a **0.034** increase), it underperformed in other domains by a quite large margin. For instance, the performance in both 'sports' and 'misc' domains dropped by **0.2027** and **0.2936**, respectively. This could potentially be due to a few reasons: (1) the 'products' domain contained fewer samples than 'screen' domain, and (2) the content in'screen' domain was inherently more similar to that of 'music, 'literature, and 'stage' domains.

Furthermore, the approach of merging together 'screen' and 'product' domains improved the model's performance: except for the 'stage' and 'sports' domains, this approach allowed to achieve superior performance in every other domain. For this reason, the combinations of these two categories were used as testing data for every following experiment.

Test domain	Screen	Products	Screen + Products
Screen	0.4842	0.3862	-
Music	0.4485	0.3946	**0.4584**
Literature	0.4158	0.3789	**0.4368**
Products	0.4515	0.5117	-
Games	0.4552	0.4836	**0.5195**
Restaurants	0.4213	0.4553	**0.4790**
Stage	**0.4602**	0.4024	0.4244
Sports	**0.2979**	0.0952	0.2925
Misc	0.5686	0.2750	**0.6627**

Table 3: Batista F1-scores after training the model on 'screen', 'product' domains, and their combination. Intersection points where train and test domains are equal contain baseline results (using 80% of data for training and 20% for testing). Merging data from different domains proved to be useful.

Test domain	Add-10	Add-100
Music	0.4588	**0.4616**
Literature	0.4313	**0.4457**
Games	0.5169	**0.5313**
Restaurants	0.4728	**0.5022**
Stage	**0.4788**	0.4453
Sports	**0.4189**	0.3353

Table 4: Batista F1-scores after adding 10 or 100 samples from target domain to the training data, initially consisting of data from 'screen' and 'product' categories.

5.3 Adding a fixed number of data from target domain

The results of the experiment found in Table 4 demonstrate improved performance in practically all domains. Moreover, adding 100 samples from target category yielded better performance in almost all domains (except for 'stage' and 'sports' domains) when compared to only adding 10 samples or using data from 'screen' and 'product' domains.

However, it appeared that utilizing 100 samples degraded performance in 'sports' domain quite significantly, when compared to the effect of using 10 samples. Given that 'sports' domain comprised a small amount of data, it is plausible to assume that after sampling 100 data instances, the few instances that were left for testing differed from the sampled data.

Test domain	0.1 ratio	0.2 ratio	0.3 ratio
Music	0.4762	0.4738	**0.4771**
Literature	0.4291	0.4555	**0.4564**
Games	0.5195	0.5238	**0.5425**
Restaurants	0.4969	0.5057	**0.5134**
Stage	0.4859	**0.5011**	0.4973
Sports	0.3317	0.3879	**0.4362**

Table 5: Batista F1-scores after taking a certain ratio of data from each domain, and adding it to the training data. Each run was conducted 3 times, using different seeds. The resulting scores are average of 3 runs.

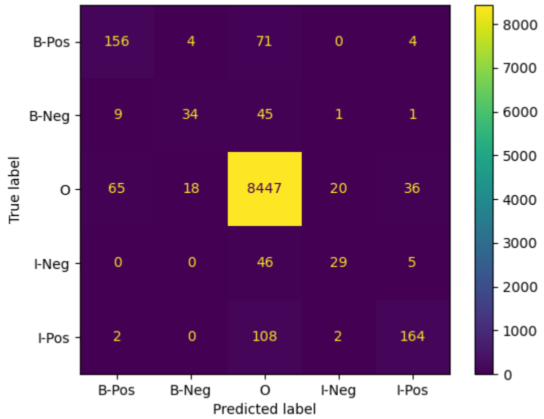

Figure 3: Confusion matrix from ’games’ domain, evaluations derived from the epoch with the highest F1-score.

5.4 Adding ratio of data from every domain

As described in Section 4, the results for this experiment were derived by running 3 experiments, utilizing various ratios of data (**0.1**, **0.2**, **0.3**) from every domain. Furthermore, each of the experiments was conducted 3 times, each time taking use of a different random seed. This ensured that various samples from each domain were employed in training and tesing. The final results, which are the average of 3 runs, are illustrated in Table 5. It appeared, that utilizing 30% of samples from each domain proved the most beneficial for the model performance.

In general, it clearly seems that adding more data from each domain and allowing the model to fine-tune on it helped the model to yield greater performance. The model trained on a combination of only ’screen’ and ’product’ domains produced inferior results as opposed to the same model trained on data with an additional 30% of samples from each domain. For a more precise comparison, the latter model yielded an increase in Batista F1-scores equal to **0.0187** in ’music’ domain, **0.0196** in ’literature’ domain, **0.023** in ’games’ domain, **0.0344** in ’restaurants domain’, **0.0729** in ’stage’ domain and **0.1437** in ’sports’ domain.

6 Error Analysis

This section describes and analyses errors the model produced by means of inspecting a confusion matrix, in addition to investigating a number of sentences and their respective predictions and gold labels. Moreover, it inspects the most common tokens wrongly classified. In particular, it focuses on surveying the ’games’ and ’sports’ domains - the ones, where the model demonstrated the highest and lowest Batista F1 scores, respectively. All the results were derived from the epochs, where the model showed the highest score.

6.1 Confusion Matrices

Confusion matrices from ’games’ and ’sports’ domains are illustrated in Figures 3 and 4, respectively.

As it can be observed, in the ’games’ domain, the model tended to confuse the **B-Pos** (beginning of the positive entity sequence) labels with **O** (outside) labels and the other way around. This can be also observed for **B-Neg** (beginning of the negative entity sequence) tags but with a significantly lesser frequency. The exact same effect can be observed for **I-Pos** (inside positive entity sequence) tags - in fact, the error rate appears to be even higher. As with **B-Neg** tags, **I-neg** (inside negative entity sequence) tags have a lesser error frequency when it comes down to confusing them with **O** tags. This effect could potentially imply that the proportion of positive sentences in ’games’ domain was greater than that of negative sentences. Furthermore, the model gave an impression of doing an arguably good job at distinguishing between positive and negative sentences: the error rate appeared to be quite low. It is the prediction of the correct sentence span that was the most difficult to accomplish.

The same tendency could be observed in ’sports’ domain. The only major difference is that the numbers are significantly smaller. This is due to the fact that ’sports’ domain comprised a small number of samples.

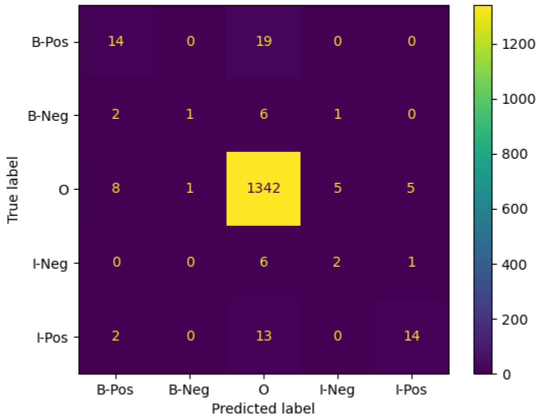

Figure 4: Confusion matrix from ’sports’ domain, evaluations derived from the epoch with the highest F1-score.

Token	Gold	Pred
Astrid	O	B-Pos
Lindgrens	O	I-Pos
figurer	B-Pos	I-Pos
gjør	O	O
seg	O	O
godt	O	O
i	O	O
et	O	O
barnespill	O	O
.	O	O

Table 6: Sentence from ’games’ domain with incorrectly predicted entity spans. English translation of the sentence is as follows: ’*Astrid Lindgren’s figures do well in a children’s game*’.

6.2 Inspecting sentences

The paper further investigated specific sentences from both domains, their respective gold labels, and the model’s predictions.

After observing a number of sentences, one could argue that the gold tags aren’t necessarily faultless. A sentence from ’games’ domain with an incorrectly predicted span can be observed in Table 6. While ’*Astrid Lindgres figurer (Astrid Lindgren’s figures)*’ and ’*figurer (figures)*’ are both entities, the predicted one is arguably a more ’precise’ entity than the gold entity. Moreover, ’games’ domain contained two following sentences: (1) ’*Jeg elsker jobben min (I love my job)*’ and (2) ’*På den står det med store bokstaver at jeg elsker jobben min (It says in big letters that I love my job)*’. Whereas the first sentence included gold positive entity ’*jobben min (my job)*’, that was not the case for the second sentence. It is debatable whether the second sentence should have had the entity omitted. Another example sentence that seemed intriguing was ’*Det er herlig, fengslende og vanedannende (It is wonderful, captivating, and addictive)*’. In this sentence, all the gold labels were equal to **O**, while the model predicted that the first word ’*Det (It)*’ is **B-Pos**. Again, it is probably plausible to assume that in the context of games, the word ’*vanedannede (addictive)’* is of positive sentiment, indicating that the sentence could possibly contain a positive entity, exactly as the model predicted.

The ’sports’ domain, while being moderate in size, simultaneously contained a fair amount of noisy sentences. One such was ’*Jone Samuelsen 3*’. The sentence appeared to be merely a name, with first and last names tagged with **B-Neg** and **I-Neg**, respectively, whereas the model predicted the same tags, but with positive polarity. In this sentence and context, the name was solely a neutral entity.

Overall, disputable suchlike sentences might have potentially degraded the model’s performance. While sentences that the model could not predict correctly because of its limited capabilities were undoubtedly present, a great part of them, as the ones mentioned above, were pure noise.

6.3 Most common error words

The top 10 wrongly classified words for both ’games’ and ’sports’ domains are illustrated in Tables 7 and 8, respectively. Correlating to the findings from confusion matrices in 6.1, it can be observed from the tables that the most common source of error was defining the correct span of an entity. This can specifically be noticed in Table 7: quotation marks ’«’ ’»’ were the most commonly misclassified tokens. Moreover, the model seemed to struggle to differentiate between all the other tags and **O** tags.

7 Conclusion

This paper analyzed the effects of conducting the task of TSA while utilizing different combinations of data from heterogenous domains, both for training and testing. By making use of the NoReC$_{\text{fine}}$ dataset, and its various categories, while training and testing NorBERT$_{3,\text{base}}$ model, it was found that adding a bigger ratio of data from each domain to train data yielded better performance in every single domain. Moreover, it was discovered that

Token	Count	Gold	Pred
«	16	B-Neg	O
»	15	I-Neg	O
spillet	12	O	B-Pos
det	8	B-Pos	O
og	7	I-Pos	O
av	6	O	I-Pos
"	5	I-Pos	O
Det	5	O	B-Pos
spill	5	O	B-Pos
3	4	O	I-pos

Table 7: Top-10 wrongly classified tokens in ’games’ domain

Token	Count	Gold	Pred
Lionel	2	B-Pos	O
Messi	2	I-Pos	O
Manuel	2	O	I-Neg
Vi	2	B-Pos	O
Higuaín	2	O	B-Neg
Neuer	2	B-Neg	B-Pos
mot	2	B-Neg	O
Götze	2	B-Pos	O
laget	2	B-Pos	O
Estland	2	I-Neg	O

Table 8: Top-10 wrongly classified tokens in ’sports’ domain

testing the model on ’games’ produced the highest F1-scores, while ’sports’ category was the hardest to predict. Additionally, the error analysis in the paper demonstrated that the model was able to distinguish between positive and negative polarities, but struggled to predict correct entity spans. It also showed that some of the sentences’ gold labels were potentially debatable, or served as a noise for the model.

8 Future work

Given the fact that NoReC$_{\text{fine}}$ comprises several domains, while only a few were more thoroughly tested in this paper, it would be intriguing to see how the combination of other domains would affect model performance. Moreover, a greater number of potentially more powerful models for Norwegian exist, offering possibilities to conduct these tests utilizing them. Additionally, it would be of great interest to implement Behavioral Testing (Ribeiro et al., 2020) for analyzing the effects of cross-domain TSA. Given the lack of data in certain domains, it would potentially prove beneficial to create short, simple sentences for each domain to investigate models’ behavior.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–

4912, Online. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023a. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Sergeevna Palatkina. 2023b. Norbench – a benchmark for norwegian language models. In *The 24rd Nordic Conference on Computational Linguistics*.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.