Exploring Cross-Domain Effects in Targeted Sentiment Analysis for Norwegian

Martin Gintovt
martigin@ifi.uio.no

Abstract

Targeted Sentiment Analysis is the task of identifying a target of an opinion and classifying its corresponding polarity in a given text. This paper analyzes the effects of training neural models for that task while utilizing data for training and testing from (partly) heterogeneous domains or categories. The results demonstrate that models trained on the "screen" domain, containing reviews about movies and TV series show adequate performance when tested on other domains, while some categories are inherently difficult. Additionally, the results illustrate 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, Targeted Sentiment Analysis (TSA) would provide entity-level sentiment for a specific target 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. TSA is a subset of SA and is a more fine-grained task. 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's aim is to investigate the cross-domain effects in the task of TSA for Norwegian. More specifically, it focuses on $\text{NoReC}_{\text{TSA}}$ dataset, 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 analyzes 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 makes use of $\text{NoReC}_{\text{TSA}}$, a dataset for fine-grained sentiment analysis in Norwegian. It is derived from $\text{NoReC}_{\text{fine}}$ dataset (Øvrelid et al., 2020), which in turn comprises a subset of the Norwegian Review Corpus (Velldal et al., 2018). The text units in the parent $\text{NoReC}_{\text{fine}}$ 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. 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'.

The data in $\text{NoReC}_{\text{TSA}}$ comes in conll-format, and includes a total of 5 labels, represented as **BIO**-labels with additional polarity: **B-targ-Positive** (B-Pos, beginning of the positive entity sequence), **I-targ-Positive** (I-Pos, inside positive entity sequence), **B-targ-Negative** (B-Neg, beginning of the negative entity sequence), **I-targ-Negative** (I-Neg, inside negative entity sequence), and **O** (outside). Figure 1 illustrates the data spread in the combination of train, development, and test splits of the dataset, whereas Table 1 demonstrates an

Token	Label
Ocarina	B-Pos
of	I-Pos
Time	I-Pos
kalles	O
ofte	O
tidenes	O
beste	O
spill	O
.	O

Table 1: Example sentence from $\text{NoReC}_{\text{TSA}}$ 'games' domain. *Ocarina of Time* is an entity, labeled with positive sentiment. English translation of the sentence is as follows: '*Ocarina of Time is often called the best game of all time.*'.

Domain	Sentences
Screen	3807
Music	2692
Literature	1089
Products	2181
Games	767
Restaurants	340
Stage	376
Sports	149
Misc	36

Table 2: Number of sentences in each domain of $\text{NoReC}_{\text{TSA}}$ dataset: test, development, and train splits combined.

example word sequence with its respective labels extracted from the 'games' domain. Additionally, the number of sentences per domain (train, development, and test data combined) can be found in 2. The data can be obtained in the GitHub repository of the IN5550 course[1].

3 Models

For the task of TSA this paper employs 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 conducts an experiment where several variants of NorBERT_3 are tested on the $\text{NoReC}_{\text{fine}}$ 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 motivate the use of the $\text{NorBERT}_{3,\text{base}}$ model, which is the main model utilized in the following experiments. The model comprises 123M parameters, 12 hidden layers, 12 attention heads, and hidden dimensions of size 786.

There is no motivation to choose this specific model for experiments - they solely require a model, powerful enough and trained on Norwegian texts. The ultimate goal is to observe how the performance differs based on the domains, not achieve state-of-the-art result.

4 Experimentation

Several experiments are carried out in this paper, where each one is conducted using the same model hyperparameters: batches of size **16**, **8** training epochs, a learning rate of **3e-5**, fine-tuning all of the models' parameters. The motivation behind the hyperparameters choice boils down to the goal of longer training with a lower learning rate. While a higher learning rate would possibly lead to a faster convergence, the aim is to see how the model performs over a longer span of time.

The first experiment described in subsection 4.1 concerns the selection of the NorBERT_3 model for further analysis. Subsection 4.2 surveys how using different domains, as well as their combination affects model performance. An investigation of the impact of adding a fixed number of samples

[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

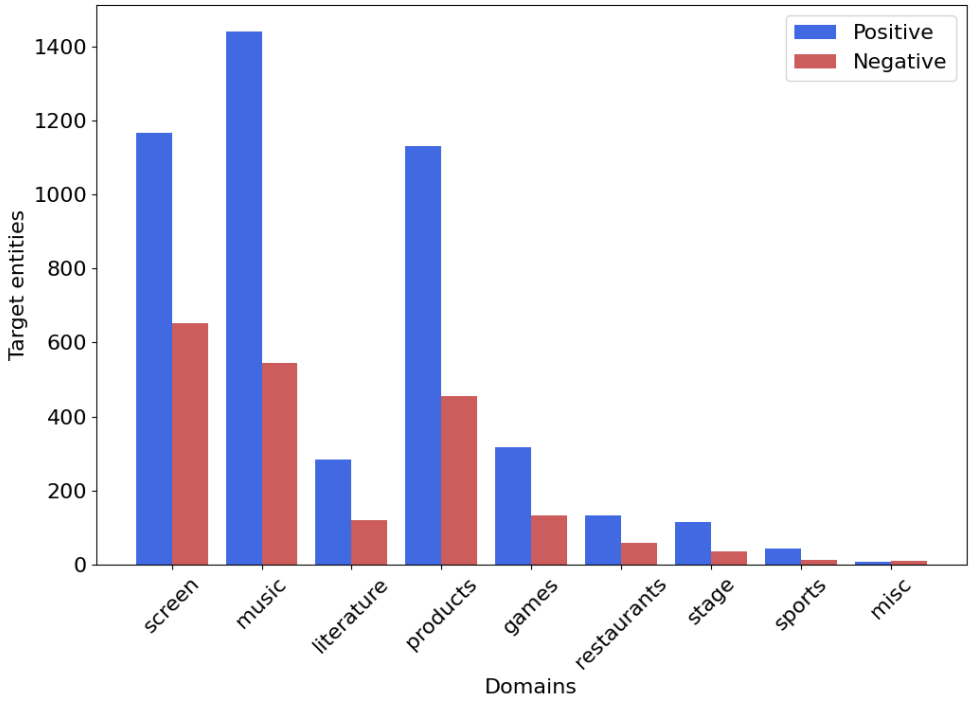

Figure 1: Spread of positively and negatively labeled target entities across domains in $\text{NoReC}_{\text{TSA}}$ data: train, development, and test splits combined. The plot was derived by counting the presence of B-Pos and B-Neg tags inside the tag sequence for each sentence, across all domains.

from target domains to training data is presented in subsection 4.3. The final experiment with adding different ratios of data from different domains is described in subsection 4.4.

4.1 Model selection

As mentioned in Section 3, several versions of NorBERT_3 are present at the moment of writing of this paper. The goal of the experiment is to determine the best-performing model to use in further analysis. In particular, 4 different versions of the model are tested: $\text{NorBERT}_{3,\text{small}}$, $\text{NorBERT}_{3,\text{base}}$, $\text{NorBERT}_{3,\text{large}}$, and $\text{NorBERT}_{3,\text{wiki-base}}$. A baseline is established for each model by training it on the train split, while assessing performance on the test split, provided by $\text{NoReC}_{\text{TSA}}$ dataset. Both splits comprise a combination of samples from nearly all domains, except 'sports' and 'misc'. These domains are therefore excluded in this experiment. Consequently, the most efficient and best-performing model is picked for further experiments. Finally, the experiment investigates the performance of the model on each separate domain from the test split.

4.2 Merging and training on different domains

The second experiment comes down to testing whether using data originating only from one or several different domains would potentially prove useful. The 'screen' and 'products' domains are of particular interest. The first one comprises the largest number of samples, whereas the latter, as described by the authors of the $\text{NoReC}_{\text{fine}}$ dataset, is perhaps the most diverse category, which comprises product reviews across a number of sub-categories. Consequently, it is intriguing to see whether such a domain will function well as the only source of training data. To conduct this and further experiments, all three data splits are merged together. Furthermore, 80% of the data from train domains is utilized for fine-tuning, while the remaining 20% is used for the purposes of testing. This is done in order to see how well will the model perform when trained and tested on data from the same domain ('screen' and 'products' in this particular case). The reason for merging the data is to increase the total amount of training and testing samples.

Furthermore, it seems intuitively plausible that combining data from different domains and using that data for training could potentially increase the

model's performance. Consequently, another experiment is conducted where the 'screen' and 'product' domains are merged together and used for training the model, while data from the remaining domains are utilized for testing purposes. Whereas the screen domain contains the most amount of data, the product domain serves as the most "universal" category.

It is important to note that during inference in the current and all subsequent experiments, a different amount of sentences is utilized in each domain. Specifically, each domain is tested using all the samples in that domain. The reasoning behind this decision boils down to some domains comprising a relatively low number of samples (e.g. sports). Utilizing an equal amount of samples would imply using only a small number of samples for testing in each domain, while there is an interest to see how the models will behave when tested on a broader amount of data.

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

In this experiment, the aim is to investigate the effect of adding a fixed amount of samples from the target domains to the training data. In particular, two tests are conducted: adding 10 and 100 samples. Furthermore, it is determined to drop the misc category in the testing phase, as it contains way too few samples for reliable performance assessment. This can be observed in Table 2.

4.4 Adding ratio of data from all domain to training data

This experiment focuses 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 are 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 are employed for each run, ensuring some randomness when sampling data for training, and resulting in 9 runs in total. Consequently, the experiment results end up being more reliable, owing to the fact that the testing subset id 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 3. The metric utilized for assessing performance is F1-score, based on the Batista algorithm[5]. The results show that NorBERT$_{3,\text{large}}$ outperformed all of the models, while NorBERT$_{3,\text{wiki-base}}$ showed the lowest score.

As illustrated in Figure 2, NorBERT$_{3,\text{base}}$ and NorBERT$_{3,\text{large}}$ mostly demonstrate the highest F1-scores during tests in other domains. While the latter model tends to yield better performance in domains like 'literature', 'music', and 'products', NorBERT$_{3,\text{base}}$ outperforms it in 'restaurants' and 'stage' domains.

Given the fact that NorBERT$_{3,\text{base}}$ contains sufficiently fewer parameters than its expanded version NorBERT$_{3,\text{large}}$ (123M vs. 353M) while yielding similar performance, the choice of the model for further experiments ends on the first one. Moreover, the NoReC$_{\text{TSA}}$ is a dataset of arguably moderate size, which motivates 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 the 'screen' and 'products' domains as training data, as well as merging together them together are illustrated in Table 4. Despite the fact that the 'products' domain is perhaps the most diverse one, the model fine-tuned on it don't manage to perform significantly better than the same model fine-tuned on data from the 'screen' domain. While the model shows better performance in the 'games' domain (a **0.0284** increase) and the 'restaurants' domain (a **0.034** increase), it underperforms in other domains by a quite large margin. For instance, the performance in both 'sports' and 'misc' domains drops by **0.2027** and **0.2936**, respectively. This could potentially be due to a few reasons: (1) the 'products' domain contained fewer samples than the 'screen' domain, and (2) the content in the 'screen' domain

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

Model	Test split	Screen	Music	Literature	Products	Games	Restaurants	Stage
$\text{NorBERT}_{3,\text{small}}$	0.4652	0.4565	0.4857	0.4054	0.4706	0.5819	0.4206	0.5889
$\text{NorBERT}_{3,\text{base}}$	0.4844	0.4719	0.5580	0.3857	0.5069	0.6197	**0.4946**	**0.6296**
$\text{NorBERT}_{3,\text{large}}$	**0.5288**	0.4752	**0.5598**	**0.4461**	**0.5740**	0.5763	0.4639	0.6111
$\text{NorBERT}_{3,\text{wiki-base}}$	0.4626	**0.5047**	0.4677	0.4022	0.4391	**0.6254**	0.3117	0.5278

Table 3: Baseline performance of 4 different NorBERT_3 models trained on train split, while tested on test split of $\text{NoReC}_{\text{TSA}}$ as well as different domains from the test split.

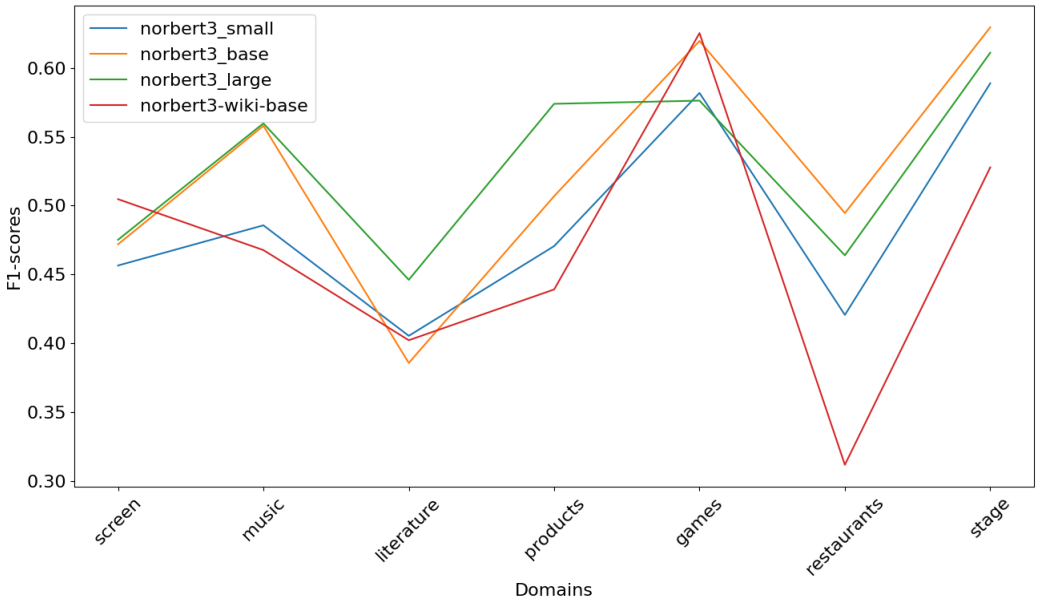

Figure 2: F1 scores achieved by each model trained on train split and tested on each domain from test split. $\text{NorBERT}_{3,\text{base}}$ and $\text{NorBERT}_{3,\text{large}}$ appear to be the most performant models.

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 4: 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).

is inherently more similar to that of 'music, 'literature, and 'stage' domains.

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

5.3 Adding a fixed number of data from target domain

The results of the experiment found in Table 5 demonstrate improved performance in practically all domains. Moreover, adding 100 samples from the target category yields better performance in almost all domains (except for the 'stage' and 'sports' domains) when compared to only adding 10 samples or using data from the 'screen' and 'product' domains.

However, it appears that utilizing 100 samples degrades performance in the 'sports' domain quite significantly when compared to the effect of using 10 samples. Given that the 'sports' domain comprises a small amount of data, it is plausible to assume that after sampling 100 data instances, the few instances that are left (49) for testing differ from the sampled data.

5.4 Adding ratio of data from every domain

As described in Section 4, the results for this experiment are derived by running 3 experiments,

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

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

utilizing various ratios of data (**0.1**, **0.2**, **0.3**) from every domain. Furthermore, each of the experiments is conducted 3 times, each time making use of a different random seed. This ensures that various samples from each domain are employed in training and testing. The final results, which are the average of 3 runs, are illustrated in Table 6. It appears, that utilizing 30% of samples from each domain proves 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 helps the model to yield greater performance. The model trained on a combination of only 'screen' and 'product' domains produces 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 yields an increase in F1-scores equal to **0.0187** in the 'music' domain, **0.0196** in the 'literature' domain, **0.023** in the 'games' domain, **0.0344** in the 'restaurants domain', **0.0729** in the 'stage' domain and **0.1437** in the 'sports' domain.

6 Error Analysis

This section describes and analyses errors the model produces 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 are derived from the epochs, where the model shows the highest score.

6.1 Confusion Matrices

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

As it can be observed, in the 'games' domain, the model tends to confuse the **B-Pos** labels with **O** labels and the other way around. This can be also observed for **B-Neg** tags but with a significantly lesser frequency. The exact same effect can be observed for **I-Pos** tags - in fact, the amount of misclassified target entities appears to be even higher. As with **B-Neg** tags, **I-neg** tags are more rarely misclassified when it comes down to confusing them with **O** tags. This effect could potentially be present due to the proportion of positive sentences in the 'games' being greater than that of negative sentences. Furthermore, the model gives an impression of doing an arguably good job at distinguishing between positive and negative sentences: not that many samples are misclassified. It is the prediction of the correct sentence span that is the most difficult to accomplish.

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

6.2 Inspecting sentences

The paper further investigates 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 the 'games' domain with an incorrectly predicted span can be observed in Table 7. While '*Astrid Lindgres figurer (Astrid Lindgren's figures)*' and '*figurer (figures)*' are both

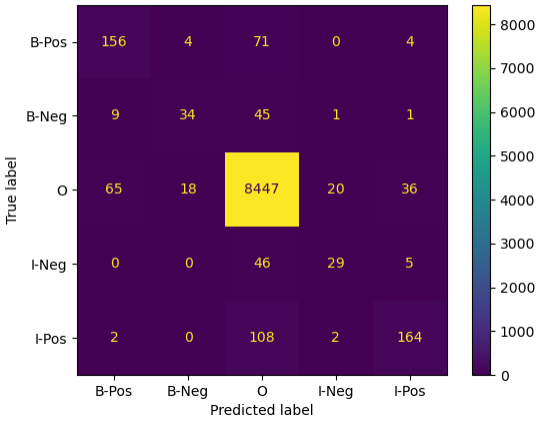

Figure 3: Confusion matrix from 'games' domain, evaluations derived from the epoch with the highest F1 score. The golden label is shown on the left, the predicted label is shown below the matrix. The diagonal of the matrix contains correctly classified target entities.

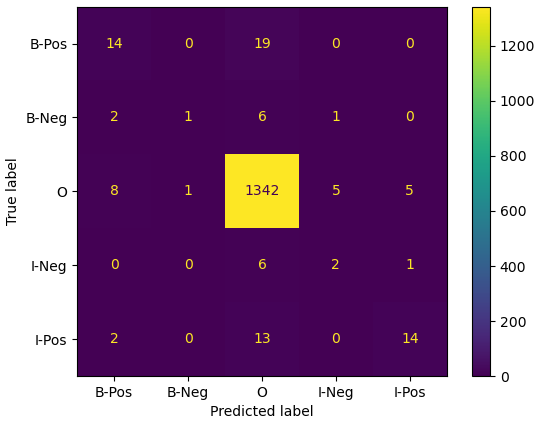

Figure 4: Confusion matrix from 'sports' domain, evaluations derived from the epoch with the highest F1 score. The golden label is shown on the left, the predicted label is shown below the matrix. The diagonal of the matrix contains correctly classified target entities.

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 7: 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*'.

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 8: Top-10 wrongly classified tokens in 'games' domain

entities, the predicted one is arguably a more 'precise' entity than the gold entity. Moreover, the 'games' domain containes 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 includes gold positive entity '*jobben min (my job)*', that is not the case for the second sentence. It is debatable whether the second sentence should have had the entity omitted. Another example sentence that seems intriguing is '*Det er herlig, fengslende og vanedannende (It is wonderful, captivating, and addictive)*'. In this sentence, all the gold labels are 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.

However, the model also demonstrates misclassifications caused by its ability to classify. An example is '*Bryce kan bytte mellom skytevåpen og sverd, et godt konsept som her dessverre har blitt miserabelt gjennomført (Bryce can switch between firearms and swords, a good concept that has unfortunately been miserably executed here)*'. Here, '*Bryce*' is given a positive polarity, whereas the gold label is negative and for the word *konsept (concept)*. Both the polarity and the span are not correct.

The 'sports' domain, while being moderate in size, simultaneously contains a fair amount of 'difficult' sentences. One such was '*Jone Samuelsen 3*'. The sentence represents a player, as well as its score. The first and last names are tagged with **B-Neg** and **I-Neg**, respectively, whereas the model predicted the same tags, but with positive polarity. It is arguably quite difficult to derive a correct prediction for this type of sentence since it is potentially ambiguous. An example of a sentence that model didn't manage to correctly classify due to its capabilities is '*- Et sensasjonelt mål av Götze! (- A sensational goal by Götze!)*'. Here, '*Götze*' is supposed to be positive, while the model didn't predict anything. This could potentially be attributed to the fact that the model have never seen the token '*Götze*' before.

Overall, the sentences in the 'sports' domain appear to be quite hard to predict correctly given their context.

6.3 Most common error words

The top 10 wrongly classified words for both 'games' and 'sports' domains are illustrated in Tables 8 and 9, 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 is defining the correct span of an entity. This can specifically be noticed in Table 8: quotation marks '«' '»' are the most commonly misclassified tokens. Moreover, the model seems to struggle to differentiate between all the other tags and **O** tags.

7 Conclusion

This paper analyzes the effects of conducting the task of TSA while utilizing different combinations of data from heterogenous domains, both for train-

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 9: Top-10 wrongly classified tokens in 'sports' domain

ing and testing. By making use of the $\text{NoReC}_{\text{fine}}$ dataset, and its various categories, while training and testing $\text{NorBERT}_{3,\text{base}}$ model, it was found that adding a bigger ratio of data from each domain to train data yield better performance in every single domain. The baseline was mostly never beaten, but that differed from domain to domain. Moreover, it was discovered that testing the model on 'games' produced the highest F1 scores, while the '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.

8 Future work

Given the fact that $\text{NoReC}_{\text{fine}}$ comprises several domains, while only a few are 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.