ScandEval: A Benchmark for Scandinavian Natural Language Understanding

Anonymous ACL submission

Abstract

This paper introduces a Scandinavian benchmarking platform, ScandEval, which can benchmark any pretrained or finetuned model on 29 datasets in Danish, Norwegian, Swedish, Icelandic and Faroese, two of which are new. We develop and release a Python package and Command-Line Interface (CLI), scandeval, which can benchmark any model that has been uploaded to the HuggingFace Hub, with reproducible results. Using this package, we benchmark over 60 Scandinavian or multilingual models and present the results of these in an interactive online leaderboard [1]. The benchmarking results shows that the investment in language technology in Norway and Sweden has led to language models that outperform multilingual models such as XLM-RoBERTa and LaBSE. We release the source code for both the package and leaderboard [2] [3].

1 Introduction

In recent years, there has been a significant increase in the number of monolingual language models in the Scandinavian languages (Møllerhøj, 2020; Højmark-Bertelsen, 2021; Sarnikowski, 2021; Enevoldsen et al., 2021; Abdaoui et al., 2020; Kummervold et al., 2021; Malmsten et al., 2020; Snæbjarnarson, 2021), to the extent that it becomes difficult both for the practioner to choose the correct model for the task at hand, as well as for language researchers to ensure that their research efforts are indeed improving upon past work.

Aside from the increasing number of models, (Sahlgren et al., 2021) also emphasises that a joint Scandinavian language model is probably a better strategy for the Scandinavian countries, considering the similarity of their languages and culture.

The languages included in the term "Scandinavian" is debatable (oxf, 2021), but we will here include Danish (da), Norwegian (nb and nn), Swedish (sv), Icelandic (is) and Faroese (fo), in line with (Sahlgren et al., 2021; Gooskens, 2020) and that these are all of the modern languages originating in Old Norse.

To help facilitate progress in both improving upon the monolingual Scandinavian models as well as the multilingual, we present ScandEval, a benchmark of Scandinavian models, along with a Python package and CLI, and an associated online leaderboard of Scandinavian language models. This benchmark covers all Scandinavian languages, in the sense described above, on a diverse set of datasets spanning named entity recognition, part-of-speech tagging, dependency parsing as well as various classification tasks.

Recent studies (Khanuja et al., 2021; Pires et al., 2019; Lauscher et al., 2020) have shown that multilingual models can outperform monolingual models when the languages are sufficiently similar, and also that they are worse than the monolingual models when the languages are too dissimilar. This shows that the Scandinavian languages could have something to gain by creating "local multilingual" models, rather than using the massively multilingual models such as XLM-RoBERTa (Conneau et al., 2020), which is why this benchmark emphasises both *Scandinavian* performance as well as the individual monolingual performance scores of the language models.

To the best of our knowledge, this is the first benchmarking tool for any of the Scandinavian languages, as well as the first online leaderboard containing scores from such a tool. Our contributions are the following:

1. We develop a Python package and CLI, scandeval, which allows reproducible benchmarking of language models on Scandinavian language datasets.

[1] (removed to preserve anonymity)

[2] https://anonymous.4open.science/r/ScandEval-5E4A

[3] (removed to preserve anonymity)

2. We benchmark all the Scandinavian and a selection of the multilingual language models on the HuggingFace Hub[4], and present all the scores in a ScandEval leaderboard on a dedicated website.

3. We uniformise all the datasets used in the benchmark, to enable consistent evaluation across languages and datasets. These uniformised datasets are also available through the above-mentioned scandeval package.

4. We provide two new sentiment analysis datasets, NoReC-IS and NoReC-FO in Icelandic and Faroese, respectively, being machine translated versions of the Norwegian Review Corpus NoReC.

2 Related Work

There have been a number of NLU benchmarks published in recent years (Wang et al., 2018; Sarlin et al., 2020; Rybak et al., 2020; Ham et al., 2020; Shavrina et al., 2020; Wilie et al., 2020; Xiang et al., 2021; Koto et al., 2020), with whom we share the same goal of advancing the state of NLP in our respective languages.

Within the Scandinavian languages specifically, the SuperLim benchmark (Adesam et al., 2020) is a Swedish NLU Benchmark featuring several difficult tasks, two of which we have included in our ScandEval benchmark: the DaLaJ and ABSAbank-Imm datasets. What distinguishes these datasets from other datasets present in SuperLim is that they contain a training split, with the other datasets in that benchmark only having a test split and are therefore not suitable for benchmarking general-purpose pretrained language models.

The XGLUE (Liang et al., 2020) dataset is another multilingual NLU benchmark. That dataset is different from ScandEval as all the training data in XGLUE is in English, and that the majority of the test sets are not available in any of the Scandinavian languages.

(Isbister and Sahlgren, 2020) features a Swedish similarity benchmark, achieved through machine translating the STS-B dataset from the GLUE benchmark (Wang et al., 2018). Aside from only dealing with a single task and a single language, the quality of the dataset is worse than a gold-standard corpus as a result of the translation, as the authors also point out.

3 Methodology

To properly evaluate the performance of a model, we ideally need to evaluate it on as many tasks as possible. Unfortunately, the Scandinavian languages do not have many openly available datasets for many downstream tasks. We thus chose to categorise the datasets that *are* available in the Scandinavian languages, and which have often been used to benchmark language models. These categories are *Named Entity Recognition* (NER), *Part-of-speech Tagging* (POS), *Dependency Parsing* (DEP) and various *Classification* tasks (CLF). We discuss the datasets used for each in Section 5.

There are two different types of models available: the *pretrained general-purpose models*, which have been trained on a large corpus, and the *finetuned models*, which are pretrained models that have been finetuned on a specific task. Evaluation of these two types of models requires different methodology, as we need to finetune the pretrained models and merely evaluate the finetuned ones.

3.1 Tokenisation

As the majority of our datasets are token classification tasks and that the language models usually use different tokenisers, we have to ensure a uniform treatment of these as well. For all token classification tasks we tokenise the documents using the pretrained tokeniser associated to the model that we are benchmarking, after which we align the resulting tokens with the gold-standard tokens.

Concretely, all gold-standard labels are propagated to the tokens they consist of. To ensure a consistent evaluation of the models we mask all but the first token in each word, ensuring that the models have to predict the same amount of labels per document.

3.2 Finetuning

The scandeval package finetunes the models for the individual tasks using the API from the transformers package (Wolf et al., 2020). The focus of the scandeval package is not to utilise the best learning algorithm for the given task, but rather to use the *same* algorithm for each model, to enable comparison between the models.

For the named entity recognition and part-of-speech tagging task, we use the AutoModelForTokenClassification class, which linearly projects the embedding from the language model encoder to each

[4] https://hf.co

token. For the classification tasks, we use the AutoModelForSequenceClassification class, which linearly projects the embedding from the language model encoder to each document.

Lastly, for the dependency parsing task, we are using the token classification class mentioned above, but as we have to predict both the head and the label of each token, we have to do this a bit differently. Firstly, we define the set of labels to be the concatenation of all the dependency parsing labels as well as the first 512 integers, the latter of which will be used to predict the head of each token. When applying the model we split the logits into the head and the label logits, and apply a softmax operation on each one individually. This then results in two labels for each token, from which we can calculate the relevant metrics.

3.3 Bootstrapping evaluation

For each finetuned model and dataset, we evaluate the model on the test dataset as well as on nine bootstrapped versions of it, generating ten scores for the (model, dataset) pair. The evaluation score is then the mean μ of these scores, along with a 95% confidence interval I_{10}, computed as

$$I_N := \mu \pm \frac{1.96}{N-1} \sum_{i=1}^{N} \texttt{score}_i. \qquad (1)$$

The pretrained models are finetuned on the training part of the given dataset, after which it is also evaluated on the test part of the dataset along with nine bootstrapped versions of the test dataset. This is repeated ten times, yielding 100 scores, and again the mean μ and 95% confidence interval I_{100} are reported.

3.4 Aggregating scores

Benchmarking a language model will result in many different scores for all the datasets in the ScandEval benchmark, and so more informative scores can be achieved through aggregation. To accomodate as many uses as possible, we aggregate in two different ways: by task and by language.

We firstly compute the *language-specific task scores* for each (model, language, task) triple, which is the mean of the scores of the model on the tasks of the language. By "task" we are here referring to the four task categories: POS, DEP, NER and CLF.

From these language-specific scores, we compute the *task score* for each (model, task) pair, as the mean of the language-specific task scores across all the languages. We also compute the *language score* for each (model, language) pair, as the mean of the language-specific task scores across all the tasks.A final ScandEval score is computed as the average of the language scores, to emphasise the training of Scandinavian models rather than monolingual ones.

3.5 Evaluation of diverging finetuned models

Evaluating finetuned models in a consistent manner is more complicated than the pretrained models, as the finetuned models might have been trained on different labels, both in terms of the names of the labels as well as which labels are included.

When it comes to the names of the labels, we include *label synonyms* in our benchmarking classes, which are simple conversion dictionaries that map the labels in the given dataset to the labels that the finetuned model has been trained on. For instance, when it comes to sentiment analysis benchmarks, the label synonyms for positive are Positive, positiv, jákvætt and LABEL_2.

Specifically for the NER models, after we have substituted the label tags according to the label synonym dictionary, any remaining label tag not present in our dictionary will be substituted for the MISC tag, to enable a fair comparison between the models.

If a model is *missing* a label, in the sense that it was not trained on a label present in the given dataset and thus that the dimension of its output projection layer is too small (a common example of this is when the model has not been trained on MISC tags), then we replace the classification head of the finetuned model with a new one, having the correct dimension, and transfer the weights from the previous finetuned head to the new one. This has the effect of the model being able to be evaluated on the dataset, but it will simply get those new labels wrong, as it will never predict them.

4 Finetuning Hyperparameters

When finetuning, we enforce a learning rate of $2 \cdot 10^{-5}$ with warmup steps equal to the size of the dataset, and a batch size of 32. If there is not enough GPU memory to finetune the model with this batch size, we halve it and double the amount of gradient accumulation, resulting in the same effective batch size. This is repeated until the batches can fit in memory. We impose a linear learning rate scheduler with intercept after 1000 epochs, and we use early stopping (Plaut et al., 1986) with a patience parameter of

$$2 + \left\lfloor \frac{1000}{\#\texttt{trainSamples}} \right\rfloor \tag{2}$$

epochs ($x \mapsto \lfloor x \rfloor$ being the floor function), to prevent models from stopping too early if the dataset is small. The choice of patience was determined qualitively, through inspection of multiple models being finetuned on multiple small datasets. The early stopping is based on performance on the validation dataset, which constitute 10% of the training dataset, sampled randomly.

We use the AdamW optimiser (Loshchilov and Hutter, 2017) with first momentum $\beta_1 = 0.9$ and second momentum $\beta_2 = 0.999$, and we optimise the cross-entropy loss throughout all tasks. Further, random seeds are fixed throughout, to ensure reproducibility.

5 ScandEval Tasks

The tasks included in this benchmark were mentioned in Section 3, and in this section we will describe the individual datasets included in the ScandEval benchmark, the modifications of them in order to benchmark the models uniformly across datasets, and what metrics are used to evaluate them.

Notably, we have enforced fixed train/test splits of the datasets. If a dataset already had such a split then we use the same test set, and combine the training and validation splits of the dataset to create a new training set. This is done for the sake of uniformity, as some of the datasets only come with preset train/test splits, while others have preset train/val/test splits.

5.1 Named Entity Recognition

For the NER task we use the four classes used in CONLL (Tjong Kim Sang and De Meulder, 2003): PER, LOC, ORG and MISC, corresponding to proper names, locations, organisations and miscellaneous entities.

In terms of evaluation metrics, we use the micro-average F1-score, which is standard for NER. We also report a *no-misc score*, which is the micro-average F1-score after we replace the MISC class in the predictions and labels with the "empty label" O. This *no-misc score* is not used in any of the aggregated scores and is purely used for comparison purposes on the individual datasets. An overview of the NER datasets used can be found in Table 1.

For Danish, we use the DaNE dataset (Hvingelby et al., 2020), being a NER tagged version of the Danish Dependency Treebank (Kromann and Lynge, 2004). DaNE is already in the CONLL format, so we perform no preprocessing on the data. It comes in a predefined train/val/test split, so we merge the training and validation datasets and keep the test dataset as-is.

For Norwegian, we use the Bokmål and Nynorsk NorNE datasets (Jørgensen et al., 2020), also being NER tagged versions of the Norwegian Dependency Treebanks (Øvrelid and Hohle, 2016). Aside from the PER, LOC, ORG and MISC tags, these also include GPE_LOC, GPE_ORG, PROD, DRV and EVT tags. We convert these to LOC, ORG, MISC, MISC and MISC, respectively. They also come in predefined train/val/test splits, so we again keep the test dataset and merge the training and validation sets.

Swedish does not have a NER tagged version of the corresponding dependency treebank, but they instead have the SUC3 dataset, a NER-enriched version of the *Stockholm-Umeå Corpus* (Gustafson-Capková and Hartmann, 2006). This dataset does not follow the CONLL format and is instead released in the XML format, with the <name> XML tags containing the NER tags for the words they span over [6]. This dataset contains the NER tags animal, event, inst, myth, other, person, place, product and work. These were converted to MISC, MISC, ORG, MISC, MISC, PER, LOC, MISC and MISC, respectively. The SUC3 dataset does not have any predefined splits, so we split the dataset at random, keeping 30% of the documents for the test dataset.

For Icelandic we use the MIM-GOLD-NER dataset (Ingólfsdóttir et al., 2020). In this NER dataset they are using the tags Person, Location, Organization, Miscellaneous, Date, Time, Money and Percent. These have been converted into PER, LOC, ORG, MISC, O, O, O and O, respectively. It comes in predefined train/val/test splits and we again merge the training and validation

[5] See https://repository.clarin.is/repository/xmlui/page/license-mim-gold-ner.

[6] The <ne> XML tags are also NER tags, but these have been automatically produced by SpaCy (Honnibal et al., 2020) models and are thus not gold standard.

Dataset	Lang	#Train	#Test	License
DaNE (Hvingelby et al., 2020)	da	4,947	565	CC-BY-SA
NorNE-NB (Jørgensen et al., 2020)	nb	18,106	1,939	CC-BY-SA
NorNE-NN (Jørgensen et al., 2020)	nn	16,064	1,511	CC-BY-SA
SUC3 (Gustafson-Capková and Hartmann, 2006)	sv	51,971	22,274	CC BY-SA
MIM-GOLD-NER (Ingólfsdóttir et al., 2020)	is	52,932	5,889	Custom[5]
WikiANN-FO (Pan et al., 2017)	fo	2,778	1,191	ODC-BY

Table 1: The NER datasets used in ScandEval.

sets. This dataset can only be used to advance linguistic research, which is in line with aims of the ScandEval benchmark.

Lastly, for Faroese we use the Faroese part of the WikiANN dataset (Pan et al., 2017). This dataset contains the NER tags PER, LOC and ORG, so these are kept as-is. This dataset does not contain predefined splits, so we split the dataset as with SUC3, keeping 30% for the test dataset.

5.2 Part-of-speech Tagging

In the part-of-speech task we used the Universal Dependencies datasets (Nivre et al., 2016), which contain the following seventeen POS tags: ADJ, ADP, ADV, AUX, CCONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB and X. No preprocessing was needed, as the Universal Dependencies datasets were all using the same labelling scheme. In terms of metrics, we use the accuracy score, as is common in POS evaluation. An overview of the POS datasets can be found in Table 2.

5.3 Dependency Parsing

The dependency parsing task also used the Universal Dependencies datasets (Nivre et al., 2016). However, the dependency parsing labels used in the different languages varied. To enforce a consistent labelling scheme we decided to only use the coarse dependency labels. For instance, we replace the label obl:arg with its coarse label obl. This results in thirty-seven coarse dependency labels, which are acl, advcl, advmod, amod, appos, aux, auxpass, case, cc, ccomp, compound, conj, cop, csubj, dep, det, discourse, dislocated, expl, fixed, flat, goeswith, iobj, list, mark, nmod, nsubj, nummod, obj, obl, orphan, parataxis, punct, reparandum, root, vocative and xcomp.

In terms of evaluation, we use the Labelled Attachment Score (LAS) as the metric. We also report the Unlabelled Attachment Score (UAS), but as with the *no-misc score* in our NER evaluation described in Section 5.1, we do not include these in our computations of the aggregated scores. An overview of the DEP datasets can be found in Table 2.

5.4 Text Classification

The text classification datasets are less uniform than the other tasks, which are primarily due to the lack of uniform datasets. This includes sentiment classification datasets as well as a mixture of classification datasets from various domains. We are using the macro-average F1 score as the metric for all these classification datasets, to accomodate an arbitrary number of classes. An overview of the CLF datasets can be found in Table 3.

For Danish, we included the sentiment classification datasets AngryTweets, TwitterSent and Europarl2 (Pauli et al., 2021), and LCC (Nielsen, 2018). AngryTweets and TwitterSent are Twitter datasets, and to comply with Twitter's Terms of Use we have fully anonymised the tweets by replacing all user mentions with @USER and all links by [LINK], as well as shuffling the tweets. The main reason for including four datasets within the same domain was that these datasets are quite small, making benchmarking results on each one of them varied, with the average being more stable. We also included the challenging hate speech dataset DKHate (Sigurbergsson and Derczynski, 2020).

In Norwegian, we included the sentiment classification dataset NoReC (Velldal et al., 2018), which is sufficiently large (double the size of the four Danish sentiment datasets combined), as well as the dialect classification dataset NorDial (Barnes et al., 2021). This dataset is in both Bokmål, Nynorsk as well as local Norwegian dialects, enforcing the capability of language models to distinguish between these.

Dataset	Lang	#Train	#Test	License
DDT (Kromann and Lynge, 2004)	da	4,947	565	CC-BY-SA
NDT-NB (Øvrelid and Hohle, 2016)	nb	18,106	1,939	CC-BY-SA
NDT-NN (Øvrelid and Hohle, 2016)	nn	16,064	1,511	CC-BY-SA
SDT (Nivre et al., 2006; Ahrenberg, 2015)	sv	4,788	1,212	CC-BY-SA
IDT (Rögnvaldsson et al., 2012; Jónsdóttir and Ingason, 2020)	is	6,144	768	CC-BY-SA
FDT (Tyers et al., 2018)	fo	1,308	300	CC-BY-SA

Table 2: The POS and DEP datasets used in ScandEval.

Dataset	Lang	#Train	#Test	License
AngryTweets (Pauli et al., 2021)	da	2,437	1,047	BSD 3-Clause
TwitterSent (Pauli et al., 2021)	da	1,010	437	BSD 3-Clause
Europarl2 (Pauli et al., 2021)	da	669	288	CC-BY-SA
LCC (Nielsen, 2018)	da	349	150	CC-BY
DKHate (Sigurbergsson and Derczynski, 2020)	da	2,960	329	CC-BY-SA
NoReC (Velldal et al., 2018)	no	9,384	1,181	CC-BY-NC
NorDial (Barnes et al., 2021)	no	954	110	CC0
ABSAbank-Imm (Rouces et al., 2020)	sv	4,346	484	CC-BY
DaLaJ (Volodina et al., 2021)	sv	8,572	888	CC-BY
NoReC-IS	is	9,384	1,181	CC-BY-NC
NoReC-FO	fo	9,384	1,181	CC-BY-NC

Table 3: The CLF datasets used in ScandEval.

As far as the author is aware, there is no trinary[7] Swedish sentiment classification dataset. However, there is a dataset containing the sentiment towards immigration, ABSAbank-Imm (Rouces et al., 2020), which we have included in the benchmark. We have also included the challenging DaLaJ (Volodina et al., 2021) dataset, set up as a correct grammar classification task. Both of these datasets are also part of the Swedish SuperLim benchmark (Adesam et al., 2020).

For Icelandic and Faroese, we were unable to find a document classification dataset. To remedy this, we created two new datasets, NoReC-IS and NoReC-FO, being machine translated versions of the Norwegian NoReC dataset, mentioned above. These two datasets were translated using the Scandinavian machine translation model from (Tiedemann and Thottingal, 2020).

6 Benchmarking Package and CLI

To enable every language researcher to benchmark their language models in a reproducible and consistent manner, we have developed a Python package called scandeval[8], which can benchmark any language model (pretrained and finetuned), available on the HuggingFace Hub[9].

The scandeval package is implemented as both a command-line interface and a Python package, which enables ease of use as both a stand-alone benchmarking tool as well as enabling integration with other Python scripts. The package follows a very *opinionated* approach to benchmarking, meaning that very few parameters can be changed. This is a deliberate design decision to enable consistent benchmarking of all models. The package follows the hyperparameter choices described in Section 4.

The scandeval package is compatible with models implemented in PyTorch (Paszke et al., 2019) and SpaCy (Honnibal et al., 2020), as long as the model is available on the HuggingFace Hub. The package is flexible in the sense that a specific model can be benchmarked on a particular dataset, but it also allows benchmarking of, say, all Icelandic models.

We can benchmark a particular model using the following terminal command:

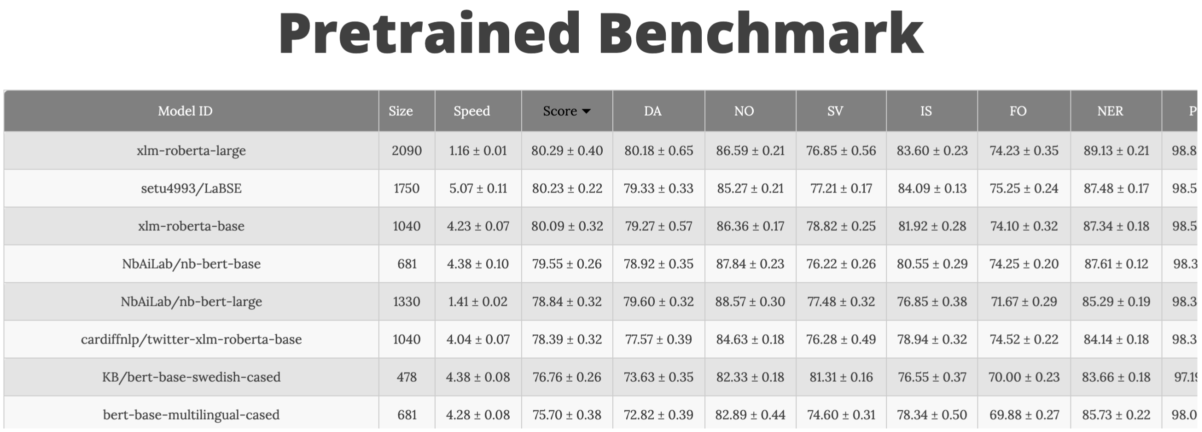
```
$ scandeval --model_id <model_id>
```

If we leave out the model_id parameter the package will benchmark all applicable models from the HuggingFace Hub, and if we specify the dataset parameter the package will only benchmark that particular dataset. The language parameter can be specified as an alternative to

[7] By "trinary" we mean a dataset containing the three classes positive, neutral and negative.

[8] https://anonymous.4open.science/r/ScandEval-5E4A

[9] https://hf.co/



Pretrained Benchmark

Model ID	Size	Speed	Score ▾	DA	NO	SV	IS	FO	NER	P
xlm-roberta-large	2090	1.16 ± 0.01	80.29 ± 0.40	80.18 ± 0.65	86.59 ± 0.21	76.85 ± 0.56	83.60 ± 0.23	74.23 ± 0.35	89.13 ± 0.21	98.8
setu4993/LaBSE	1750	5.07 ± 0.11	80.23 ± 0.22	79.33 ± 0.33	85.27 ± 0.21	77.21 ± 0.17	84.09 ± 0.13	75.25 ± 0.24	87.48 ± 0.17	98.5
xlm-roberta-base	1040	4.23 ± 0.07	80.09 ± 0.32	79.27 ± 0.57	86.36 ± 0.17	78.82 ± 0.25	81.92 ± 0.28	74.10 ± 0.32	87.34 ± 0.18	98.5
NbAiLab/nb-bert-base	681	4.38 ± 0.10	79.55 ± 0.26	78.92 ± 0.35	87.84 ± 0.23	76.22 ± 0.26	80.55 ± 0.29	74.25 ± 0.20	87.61 ± 0.12	98.3
NbAiLab/nb-bert-large	1330	1.41 ± 0.02	78.84 ± 0.32	79.60 ± 0.32	88.57 ± 0.30	77.48 ± 0.32	76.85 ± 0.38	71.67 ± 0.29	85.29 ± 0.19	98.3
cardiffnlp/twitter-xlm-roberta-base	1040	4.04 ± 0.07	78.39 ± 0.32	77.57 ± 0.39	84.63 ± 0.18	76.28 ± 0.49	78.94 ± 0.32	74.52 ± 0.22	84.14 ± 0.18	98.3
KB/bert-base-swedish-cased	478	4.38 ± 0.08	76.76 ± 0.26	73.63 ± 0.35	82.33 ± 0.18	81.31 ± 0.16	76.55 ± 0.37	70.00 ± 0.23	83.66 ± 0.18	97.1
bert-base-multilingual-cased	681	4.28 ± 0.08	75.70 ± 0.38	72.82 ± 0.39	82.89 ± 0.44	74.60 ± 0.31	78.34 ± 0.50	69.88 ± 0.27	85.73 ± 0.22	98.0

Figure 1: A sample of the online leaderboard of pretrained Scandinavian models.

Language	Top1	Top2	Top3
Overall	`xlm-roberta-large`	`setu4993/LaBSE`	`xlm-roberta-base`
Danish	`xlm-roberta-large`	`NbAiLab/nb-bert-large`	`setu4993/LaBSE`
Norwegian	`NbAiLab/nb-bert-large`	`NbAiLab/nb-bert-base`	`xlm-roberta-large`
Swedish	`KB/bert-base-swedish-cased`	`xlm-roberta-base`	`NbAiLab/nb-bert-large`
Icelandic	`setu4993/LaBSE`	`xlm-roberta-large`	`vesteinn/IceBERT`
Faroese	`setu4993/LaBSE`	`cardiffnlp/twitter-xlm-roberta-base`	`NbAiLab/nb-bert-base`

Table 4: The three best performing pretrained models in each of the language categories.

`model_id` to benchmark all models in that language. Further, all of these parameters can be specified multiple times, to benchmark all the combinations of models/languages and datasets.

Aside from its primary benchmarking capabilities, the `scandeval` package can also be used to load any of the benchmarking datasets, to enable the finetuning of new models in a way that is consistent with the train/test splits used in the benchmark. The datasets can be loaded using the `load_dataset` function; see more in the `scandeval` documentation[10].

7 Online Leaderboard

Using the `scandeval` package, we have benchmarked 40 pretrained models and 26 finetuned models in the Scandinavian languages which were available on the HuggingFace Hub. Aside from these models we also included several multilingual models to enable a fair comparison. The multilingual models included are `mBERT` (both the `base` (Devlin et al., 2019) and `distilled` (Sanh et al., 2019) versions), `XLM-RoBERTa` (Conneau et al., 2020) (both the `base` and `large` versions), the `Twitter-XLM-RoBERTa` model (Barbieri et al., 2021) and the `LaBSE` model (Feng et al., 2020). Lastly, to enable better interpretability of the results, we also benchmark a randomly initialised RoBERTa-base model on the datasets, which will make it more transparent how much "external knowledge" the pretrained models are able to utilise in their predictions. Benchmarking all these models approximately required 3,000 GPU hours on a GeForce RTX 2080 Ti.

Aside from the predictive performance of the models we also benchmarked the inference speed of the finetuned models using the `pyinfer` (Pierse, 2020), and report the mean and standard deviation of the number of inferences per second. These have been computed using a single Tesla P100-PCIE-16GB GPU, by recording the inference time of running a document with 390 characters[11] through the model one hundred times. Lastly, we also record the size of the pretrained model, measured in megabytes. These two metrics (inference speed and model size) are useful to practioners using the models, as there is often a trade-off between the accuracy and speed of the predictions. We include neither the size nor inference speed of the model in our aggregated scores, however.

We have presented all of the benchmarked results along with their associated confidence intervals in two online leaderboards: one for the pretrained

[10] `https://anonymous.4open.science/r/ScandEval-5E4A`

[11] The document in question is "Dette er en helt vild og ret lang test.", repeated ten times.

models and one for the finetuned models.[12]. These scores have been computed as described in Section 3, along with the task-specific scores, the language-specific scores and the final ScandEval score. A screenshot of the leaderboard can be seen in Figure 1.

At the time of writing, the top-3 models in terms of their ScandEval score are the XLM-RoBERTa-large model (Conneau et al., 2020), the LaBSE model (Feng et al., 2020) and the XLM-RoBERTa-base model (Conneau et al., 2020). The best models for the individual languages can be seen in Table 4.

8 Limitations and Risks

The ScandEval benchmark is currently limited to classification tasks and will therefore not be able to measure the full performance of the language models in question. This is primarily due to a lack of gold standard labelled datasets in more challenging language tasks like question-answering and summarisation. Nevertheless, we see this as sufficient for comparing the current state-of-the-art language models in the Scandinavian languages, and as such datasets become available we will be able to extend the ScandEval benchmark to accomodate these.

This focus on classification tasks could also pose a risk, favouring the development of language models which solely lead to good classifiers rather than more general-purpose language models. We have tried to minimise this risk by including a diverse mixture of both syntactic and semantic classification tasks.

9 Discussion

We note that the results presented in our online leaderboards described in Table 4 show that the efforts the National Libraries in Norway and Sweden have paid off, in the sense that their models NbAiLab/nb-bert-large (Kummervold et al., 2021) and KB/bert-base-swedish-cased (Malmsten et al., 2020) are beating the multilingual models, and in Icelandic the vesteinn/IceBERT model (Snæbjarnarson, 2021) is catching up with them.

This shows that investing in language techonologies at a national level can be worthwhile. We also see from the same table that the Norwegian model is within the top-3 best models in Norwegian, Swedish, Danish and Faroese, indicating a potential large amount of language transfer, which indicates that a joint Scandinavian approach could improve the results of the current monolingual models within the Scandinavian languages.

10 Conclusion

In this paper we have presented a benchmarking framework for the Scandinavian languages, together with a Python package and CLI, scandeval, which can be used to benchmark any model available on the HuggingFace Hub. We have also presented two online leaderboards: one for pretrained models and one for finetuned models. The results from the pretrained leaderboard show that the monolingual models trained can exceed the performance of the current state-of-the-art multilingual models, and indicate signs of a potential strong language transfer between the Scandinavian languages.

[12] These leaderboards are available at (removed to preserve anonymity).

References

2021. *Oxford English Dictionary*. Oxford University Press. https://www.lexico.com/definition/scandinavia.

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Yvonne Adesam, Aleksandrs Berdicevskis, and Felix Morger. 2020. Swedishglue–towards a swedish test set for evaluating natural language understanding models.

Lars Ahrenberg. 2015. Converting an English-Swedish parallel treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 10–19, Uppsala, Sweden. Uppsala University, Uppsala, Sweden.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2021. Xlm-t: A multilingual language model toolkit for twitter. *arXiv preprint arXiv:2104.12250*.

Jeremy Barnes, Petter Mæhlum, and Samia Touileb. 2021. NorDial: A preliminary corpus of written Norwegian dialect use. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 445–451, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Kenneth Enevoldsen, Lasse Hansen, and Kristoffer Nielbo. 2021. Dacy: A unified framework for danish nlp. *arXiv preprint arXiv:2107.05295*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Charlotte Gooskens. 2020. *The North Germanic Dialect Continuum*, Cambridge Handbooks in Language and Linguistics, page 761–782. Cambridge University Press.

Sofia Gustafson-Capková and Britt Hartmann. 2006. Manual of the stockholm umeå corpus version 2.0. *Unpublished Work*.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS: New benchmark datasets for Korean natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeglem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Rasmus Hvingelby, Amalie Brogaard Pauli, Maria Barrett, Christina Rosted, Lasse Malm Lidegaard, and Anders Søgaard. 2020. DaNE: A named entity resource for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4597–4604, Marseille, France. European Language Resources Association.

Malte Højmark-Bertelsen. 2021. Ælæctra - a step towards more efficient danish natural language processing. `https://github.com/MalteHB/-l-ctra/`.

Svanhvít L Ingólfsdóttir, Ásmundur A Gudjónsson, and Hrafn Loftsson. 2020. Named entity recognition for icelandic: Annotated corpus and models. In *International Conference on Statistical Language and Speech Processing*, pages 46–57. Springer.

Tim Isbister and Magnus Sahlgren. 2020. Why not simply translate? a first swedish evaluation benchmark for semantic similarity. *arXiv preprint arXiv:2009.03116*.

Hildur Jónsdóttir and Anton Karl Ingason. 2020. Creating a parallel Icelandic dependency treebank from raw text to Universal Dependencies. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2924–2931, Marseille, France. European Language Resources Association.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating named entities for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.

Simran Khanuja, Melvin Johnson, and Partha Talukdar. 2021. MergeDistill: Merging language models using pre-trained distillation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2874–2887, Online. Association for Computational Linguistics.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. IndoLEM and IndoBERT: A benchmark dataset and pre-trained language model for Indonesian NLP. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 757–770, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthias Trautner Kromann and Stine Kern Lynge. 2004. The danish dependency treebank v. 1.0.

Per E Kummervold, Javier De la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a Norwegian transformer model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 20–29, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden – making a swedish bert.

Jens Dahl Møllerhøj. 2020. Nordic bert. https://github.com/certainlyio/nordic_bert.

Finn Årup Nielsen. 2018. Danish resources. http://www.imm.dtu.dk/~faan/ps/Nielsen2016Danish.pdf.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585, Portorož, Slovenia. European Language Resources Association (ELRA).

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Amalie Brogaard Pauli, Maria Barrett, Ophélie Lacroix, and Rasmus Hvingelby. 2021. DaNLP: An open-source toolkit for Danish natural language processing. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 460–466, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Charles Pierse. 2020. Pyinfer. https://github.com/cdpierse/pyinfer.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

David C Plaut et al. 1986. Experiments on learning by back propagation.

Eiríkur Rögnvaldsson, Anton Karl Ingason, Einar Freyr Sigurðsson, and Joel Wallenberg. 2012. The Icelandic parsed historical corpus (IcePaHC). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1977–1984, Istanbul, Turkey. European Language Resources Association (ELRA).

Jacobo Rouces, Lars Borin, and Nina Tahmasebi. 2020. Creating an annotated corpus for aspect-based sentiment analysis in swedish. In *DHN*, pages 318–324.

Piotr Rybak, Robert Mroczkowski, Janusz Tracz, and Ireneusz Gawlik. 2020. KLEJ: Comprehensive benchmark for Polish language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1191–1201, Online. Association for Computational Linguistics.

Magnus Sahlgren, Fredrik Carlsson, Fredrik Olsson, and Love Börjeson. 2021. It's basically the same language anyway: the case for a nordic language model. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 367–372, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947.

Philip Tamimi Sarnikowski. 2021. Danish transformers. GitHub. https://github.com/sarnikowski.

Tatiana Shavrina, Alena Fenogenova, Emelyanov Anton, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4717–4726, Online. Association for Computational Linguistics.

Gudbjartur Ingi Sigurbergsson and Leon Derczynski. 2020. Offensive language and hate speech detection for Danish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3498–3508, Marseille, France. European Language Resources Association.

Vésteinn Snæbjarnarson. 2021. Automated methods for question-answering in icelandic. Master's thesis, University of Iceland, Reykjavík.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Francis Tyers, Mariya Sheyanova, Aleksandra Martynova, Pavel Stepachev, and Konstantin Vinogorodskiy. 2018. Multi-source synthetic treebank creation for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for 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).

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021. DaLAJ – a dataset for linguistic acceptability judgments for Swedish. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 28–37, Online. LiU Electronic Press.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, and Ayu Purwarianti. 2020. IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 843–857, Suzhou, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.