Large-Scale Contextualised Language Modelling for Norwegian

Andrey Kutuzov, Jeremy Barnes, Erik Velldal,
Lilja Øvrelid and Stephan Oepen

University of Oslo
Department of Informatics
Language Technology Group

{andreku|jeremycb|erikve|liljao|oe}@ifi.uio.no

Abstract

We present the ongoing NorLM initiative to support the creation and use of very large contextualised language models for Norwegian (and in principle other Nordic languages), including a ready-to-use software environment, as well as an experience report for data preparation and training. This paper introduces the first large-scale monolingual language models for Norwegian, based on both the ELMo and BERT frameworks. In addition to detailing the training process, we present contrastive benchmark results on a suite of NLP tasks for Norwegian.

For additional background and access to the data, models, and software, please see:

http://norlm.nlpl.eu

1 Introduction

In this work, we present *NorLM*, an ongoing community initiative and emerging collection of large-scale contextualised language models for Norwegian. We here introduce the NorELMo and NorBERT models, that have been trained on around two billion tokens of running Norwegian text. We describe the training procedure and compare these models with the multilingual mBERT model (Devlin et al., 2019), as well as an additional Norwegian BERT model developed contemporaneously, with some interesting differences in training data and setup. We report results over a number of Norwegian benchmark datasets, addressing a broad range of diverse NLP tasks: part-of-speech tagging, negation resolution, sentence-level and fine-grained sentiment analysis and named entity recognition (NER).

All the models are publicly available for download from the Nordic Language Processing Laboratory (NLPL) Vectors Repository[1] with a CC BY 4.0 license. They are also accessible locally, together with the training and supporting software, on the two national superclusters Puhti and Saga, in Finland and Norway, respectively, which are available to university NLP research groups in Northern Europe through the Nordic Language Processing Laboratory (NLPL).[2] The NorBERT model is in addition served via the Huggingface Transformers model hub.[3]

NorLM is a joint effort of the projects EOSC-Nordic (European Open Science Cloud) and SANT (Sentiment Analysis for Norwegian), coordinated by the Language Technology Group (LTG) at the University of Oslo. The goal of this work is to provide these models and supporting tools for researchers and developers in Natural Language Processing (NLP) for the Norwegian language. We do so in the hope of facilitating scientific experimentation with and practical applications of state-of-the-art NLP architectures, as well as to enable others to develop their own large-scale models, for example for domain- or application-specific tasks, language variants, or even other languages than Norwegian. Under the auspices of the NLPL use case in EOSC-Nordic, we are also coordinating with colleagues in Denmark, Finland, and Sweden on a collection of large contextualised language models for the Nordic languages, including language variants or related groups of languages, as linguistically or technologically appropriate.

2 Background

Bokmål and Nynorsk There are two official standards for written Norwegian; *Bokmål*, the main variety, and *Nynorsk*, used by 10–15% of

[1] http://vectors.nlpl.eu/repository
[2] http://www.nlpl.eu
[3] https://huggingface.co/ltgoslo/norbert

the Norwegian population. Norwegian language legislation specifies that minimally 25% of the written public service information should be in Nynorsk. While the two varieties are closely related, there can also be relatively large differences lexically (though often with a large degree of overlap on the character-level still). Several previous studies have indicated that joint modeling of Bokmål and Nynorsk works well for many NLP tasks, like tagging and parsing (Velldal et al., 2017) and NER (Jørgensen et al., 2020). The contextualised language models presented in this paper are therefore trained jointly on both varieties, but with the minority variant Nynorsk represented by comparatively less data than Bokmål (reflecting the natural usage).

Datasets For all our models presented below, we used the following training corpora:

1. Norsk Aviskorpus (NAK), a collection of Norwegian news texts[4] (both Bokmål and Nynorsk) from 1998 to 2019; 1.7 billion words;

2. Bokmål Wikipedia dump from September 2020; 160 million words;

3. Nynorsk Wikipedia dump from September 2020; 40 million words.

The corpora contain ordered sentences (which is important for BERT-like models, because one of their training tasks is next sentence prediction). In total, our training corpus comprises about two billion (1,907,072,909) word tokens in 203 million (202,802,665) sentences.

We conducted the following pre-processing steps:

1. Wikipedia texts were extracted from the dumps using the `segment_wiki` script from the Gensim project (Řehůřek and Sojka, 2010).

2. For the news texts from Norwegian Aviskorpus, we performed de-tokenization and conversion to UTF-8 encoding, where required.

3. The resulting corpus was sentence-segmented using Stanza (Qi et al., 2020). We left blank lines between documents (and sections in the case of Wikipedia) so that the 'next sentence prediction' task of BERT does not span between documents.

3 Prerequisites: software and computing

Developing very large contextualised language models is no small challenge, both in terms of engineering sophistication and computing demands. Training ELMo- and in particular BERT-like models presupposes access to specialised hardware – graphical processing units (GPUs) – over extended periods of time. Compared to the original work at Google or to our sister initiative at the National Library of Norway (see below), our two billion tokens in Norwegian training data can be characterised as moderate in size.

Nevertheless, training a single NorBERT model requires close to one full year of GPU utilisation, which through parallelization over multiple compute nodes, each featuring four GPUs, could be completed in about three weeks of wall clock time. At this scale, premium software efficiency and effective parallelization are prerequisites, not only to allow repeated incremental training and evaluation cycles to complete in practical intervals, but equally so for cost-efficient utilisation of scarce, shared computing resources and, ultimately, a shred of environmental sustainability.

To prepare the NorLM software environment, we have teamed up with support staff at the Norwegian national e-infrastructure provider, Uninett Sigma2, and developed a fully automated and modularised installation procedure using the EasyBuild framework (`https://easybuild.io`). All necessary tools are compiled from source with the right set of hardware-specific optimizations and platform-specific optimised libraries for basic linear algebra ('math kernels') and communication across multiple compute nodes.

This approach to software provisioning makes it possible to (largely) automatically create fully parallel training and experimentation environments on multiple computing infrastructures – in our work to date two national HPC superclusters, in Norway and Finland, but in principle just as much any suitable local GPU cluster. In our view, making available both a ready-to-run software environment on Nordic national e-infrastructures, where university research groups typically can gain no-cost access, coupled with the recipe for recreating the environment on other HPC systems, may

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

contribute to 'democratising' large-scale NLP research; if nothing else, it eliminates dependency on commercial cloud computing services.

4 Related work

Large-scale deep learning language models (LM) are important components of current NLP systems. They are often based on BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) and other contextualised architectures. A number of language-specific initiatives have in recent years released monolingual versions of these models for a number of languages (Fares et al., 2017; Kutuzov and Kuzmenko, 2017; Virtanen et al., 2019; de Vries et al., 2019; Ulčar and Robnik-Šikonja, 2020; Koutsikakis et al., 2020; Nguyen and Nguyen, 2020; Farahani et al., 2020; Malmsten et al., 2020). For our purposes, the most important such previous training effort is that of Virtanen et al. (2019) on creating a BERT model for Finnish – FinBERT[5] – as our training setup for creating NorBERT builds heavily on this; see Section 6 for more details.

Many low-resource languages do not have dedicated monolingual large-scale language models, and instead resort to using a multilingual model, such as Google's multilingual BERT model – mBERT – which was trained on data that also included Norwegian. Up until the release of the models described in the current paper, mBERT was the only BERT-instance that could be used for Norwegian.[6]

Another widely used architecture for contextualised LMs is Embeddings From Language Models or ELMo (Peters et al., 2018). The *ElmoForManyLangs* initiative (Che et al., 2018) trained and released monolingual ELMo models for a wide range of different languages, including Norwegian (with separate models for Bokmål and Nynorsk). However, these models were trained on very modestly sized corpora of 20 million words for each language (randomly sampled from Wikipedia dumps and Common Crawl data).

In a parallel effort to that of the current paper, the AI Lab of the National Library of Norway, through their Norwegian Transformer Model (NoTraM) project, has released a Norwegian BERT (Base, cased) model dubbed NB-BERT (Kummervold et al., 2021).[7] The model is trained on the Colossal Norwegian Corpus, reported to comprise close to 18,5 billion words (109.1 GB of text).

In raw numbers, this is about ten times more than the corpus we use for training the NorLM models. However, the vast majority of this is from OCR'ed historical sources, which is bound to introduce at least some noise. In Section 7 below, we demonstrate that in some NLP tasks, a language model trained on less (but arguably cleaner) data can outperform a model trained on larger but noisy corpora.

5 NorELMo

NorELMo is a set of bidirectional recurrent ELMo language models trained from scratch on the Norwegian corpus described in Section 1. They can be used as a source of contextualised token representations for various Norwegian natural language processing tasks. As we show below, in many cases, they present a viable alternative to Transformer-based models like BERT. Their performance is often only marginally lower, while the compute time required to adapt the model to the task at hand can be an order of magnitude less on identical hardware.

Currently we present two models, with more following in the future:

1. **NorELMo**$_{30}$: 30,000 most frequent words in the vocabulary

2. **NorELMo**$_{100}$: 100,000 most frequent words in the vocabulary

Note that independent of the vocabulary size, both NorELMo$_{30}$ and NorELMo$_{100}$ can process arbitrary word tokens, due to the ELMo architecture (where the first CNN layer converts input strings to non-contextual word embeddings). Thus, the size of the vocabulary controls only the number of words used as targets for the language modelling task in the course of training. Supposedly, the model with a larger vocabulary is more effective in treating less frequent words at the cost of being less effective with more frequent words.

Each model was trained for 3 epochs with batch size 192. We employed a version of the original

[5] https://github.com/TurkuNLP/FinBERT

[6] A BERT model trained on Norwegian data was published at https://github.com/botxo/nordic_bert in the beginning of 2020. However, the vocabulary of this model seems to be broken, and to the best of our knowledge nobody has achieved any meaningful results with it.

[7] https://github.com/NBAiLab/notram

ELMo training code from Peters et al. (2018) updated to work better with the recent TensorFlow versions. All the hyperparameters were left at their default values, except the LSTM dimensionality reduced to 2,048 from the default 4,096 (in our experience, this rarely influences performance). Training of each model took about 100 hours on four NVIDIA P100 GPUs.

These are the first ELMo models for Norwegian trained on a large corpus. As has already been mentioned, the Norwegian ELMo models from the *ElmoForManyLangs* project (Che et al., 2018) were trained on very small corpora samples and seriously under-perform on semantic-related NLP tasks, although they can yield impressive results on POS tagging and syntactic parsing (Zeman et al., 2018). In addition, they were trained with custom code modifications and can be used only with the custom *ElmoForManyLangs* library. On the other hand, our NorELMo models are fully compatible both with the original ELMo implementation by Peters et al. (2018) and with the more modern *simple_elmo* Python library provided by us.[8]

The vocabularies are published together with the models. For different tasks, different models can be better, as we show below. The published packages contain both TensorFlow checkpoints (for possible fine-tuning, if need be) and model files in the standard Hierarchical Data Format (HDF5) for easier inference usage. In addition, we have setup ELMoViz, a demo web service to explore Norwegian ELMo models.[9]

6 NorBERT

Our NorBERT model is trained from scratch for Norwegian, and can be used in exactly the same way as any other BERT-like model. The NorBERT training setup heavily builds on prior work on FinBERT conducted at the University of Turku (Virtanen et al., 2019).

NorBERT features a custom WordPiece vocabulary which is case-sensitive and includes accented characters. It has much better coverage of Norwegian words than the mBERT model or NB-BERT (which uses the same vocabulary as mBERT). This is clearly seen on the example of the tokenization performed by both for the Norwegian sentence '*Denne gjengen håper at de sammen skal bidra til å gi kvinnefotballen i Kristiansand et lenge etterlengtet løft*'

- **mBERT/NB-BERT:** 'Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft'
- **NorBERT:** 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterl ##engt ##et løft'

NorBERT tokenization splits the sentence into pieces which much better reflect the real Norwegian words and morphemes (cf. '*k vinne fo t ball en*' versus '*kvinne fotball en*'). We believe this to be extremely important for more linguistically-oriented studies, where it is critical to deal with words, not with arbitrarily fragmented pieces (even if they are well-performing in practical tasks).

The vocabulary for the model is of size 30,000. It is much less than the 120,000 of mBERT, but it is compensated by these entities being almost exclusively Norwegian. The vocabulary was generated from raw text, without, e.g., separating punctuation from word tokens. This means one can feed raw text into NorBERT.

For the vocabulary generation, we used the SentencePiece algorithm (Kudo, 2018) and Tokenizers library.[10] The resulting Tokenizers model was converted to the standard BERT WordPiece format. The final vocabulary contains several thousand unused wordpiece slots which can be filled in with task-specific lexical entries for further fine-tuning by future NorBERT users.

6.1 Training technicalities

NorBERT corresponds in its configuration to the Google's Bert-Base Cased for English, with 12 layers and hidden size 768 (Devlin et al., 2019). We used the standard masked language modeling and next sentence prediction losses with the LAMB optimizer (You et al., 2020). The model was trained on the Norwegian academic HPC system called Saga. Most of the time the training process was distributed across 4 compute nodes and 16 NVIDIA P100 GPUs. Overall, it took approximately 3 weeks (more than 500 hours).

[8] https://pypi.org/project/simple-elmo/

[9] http://vectors.nlpl.eu/explore/embeddings/en/contextual/

[10] https://github.com/huggingface/tokenizers

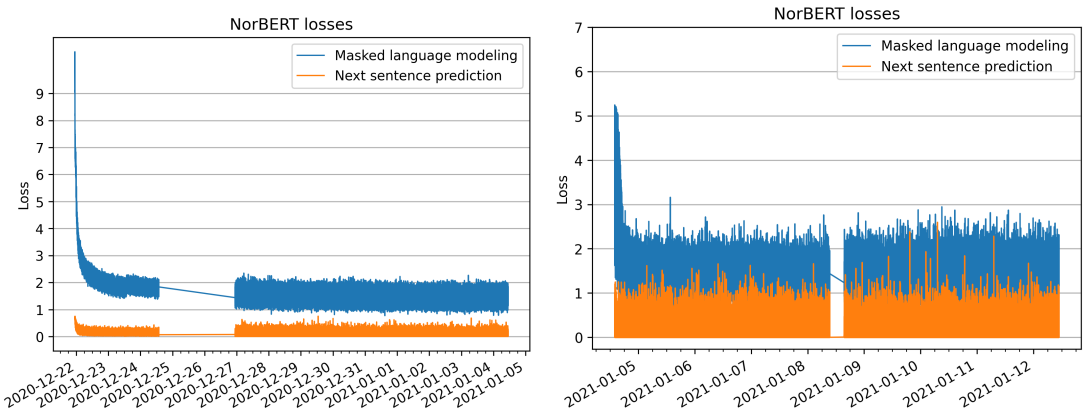

Figure 1: NorBERT loss plots at the Phase 1 (left) and Phase 2 (right).

Similar to Virtanen et al. (2019), we employed the BERT implementation by NVIDIA[11], which allows fast multi-node and multi-GPU training.

We made minor changes to this code, mostly to adapt it to the newer TensorFlow versions. All these patches and the utilities we used at the preprocessing, training and evaluation stages are published in our GitHub repository.[12] Instructions to reproduce the training setup with the EasyBuild software build and installation framework are also available.[13]

6.2 Training workflow

Phase 1 (training with maximum sequence length of 128) was done with batch size 48 and global batch size 48*16=768. Since one global batch contains 768 sentences, approximately 265,000 training steps constitute 1 epoch (one pass over the whole corpus). We have done 3 epochs: 795,000 training steps.

Phase 2 (training with maximum sequence length of 512) was done with batch size 8 and global batch size 8*16=128. We aimed at mimicking the original BERT in that at Phase 2 the model should see about 1/9 of the number of sentences seen during Phase 1. Thus, we needed about 68 million sentences, which at the global batch size of 128 boils down to 531,000 training steps more.

The loss plots are shown in Figure 1 (the training was on pause on December 25 and 26, since we were solving problems with mixed precision training). Full logs are available at the GitHub repository.

[11] https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/LanguageModeling/BERT, version 20.06.08

[12] https://github.com/ltgoslo/NorBERT

[13] http://wiki.nlpl.eu/index.php/Eosc/pretraining/nvidia

Task	Train	Dev	Test
POS Bokmål	15,696	2,409	1,939
POS Nynorsk	14,174	1,890	1,511
NER Bokmål	15,696	2,409	1,939
NER Nynorsk	14,174	1,890	1,511
Sentence-level SA	2,675	516	417
Fine-grained SA	8,543	1,531	1,272
Negation	8,543	1,531	1,272

Table 1: Number of sentences in the training, development, and test splits in the datasets used for the evaluation tasks.

7 Evaluation

This section presents benchmark results across a range of different tasks. We compare NorELMO and NorBERT to both mBERT and to the recently released NB-BERT model described in Section 4. Where applicable, we show separate evaluation results for Bokmål and Nynorsk. Below we first provide an overview of the different tasks and the corresponding classifiers that we train, before turning to discuss the results.

7.1 Task descriptions

We start by briefly describing each task and associated dataset, in addition to the architectures we use. The sentence counts for the different datasets and their train, dev. and test splits are provided in Table 1.

Part-of-speech tagging The Norwegian Dependency Treebank (NDT) (Solberg et al., 2014) in-

cludes annotation of POS tags for both Bokmål and Nynorsk. NDT has also been converted to the Universal Dependencies format (Øvrelid and Hohle, 2016; Velldal et al., 2017) and this is the version we are using here (for UD 2.7) for predicting UPOS tags.

We use a typical sequence labelling approach with the BERT models, adding a linear layer after the final token representations and taking the softmax to get token predictions. We fine-tune all parameters for 20 epochs, using a learning rate of 2e-5, a training batch size of 8, max length of 256, and keep the best model on the development set. ELMo models were not fine-tuned, following the recommendations from Peters et al. (2019). Instead we trained a simple neural classifier (a feed forward network with one hidden layer of size 128, ReLU non-linear activation function and dropout), using ELMo token embeddings as features. The random seed has been kept fixed all the time. Models are evaluated on accuracy.

Named entity recognition The NorNE[14] dataset annotates the UD-version of NDT with a rich set of entity types (Jørgensen et al., 2020). The evaluation metrics here is 'strict' micro F_1, requiring both the correct entity type and exact match of boundary surface string. We predict 8 entity types: Person (PER), Organisation (ORG), Location (LOC), Geo-political entity, with a locative sense (GPE-LOC), Geo-political entity, with an organisation sense (GPE-ORG), Product (PROD), Event (EVT), Nominals derived from names (DRV). The evaluation is done using the code for the SemEval'13 Task 9[15].

We cast the named entity recognition problem as a sequence labelling task, using a BIO label encoding. For the BERT-based models, we solve it by fine-tuning the pre-trained model on the NorNE dataset for 20 epochs with early stopping and batch size 32. The resulting model is applied to the test set.

For ELMo models, we infer contextualised token embeddings (averaged representations across all 3 layers) for all words. Then, these token embeddings are fed to a neural classifier with dropout, identical to the one we used for POS tagging earlier. This classifier is also trained for 20 epochs with early stopping and batch size 32.

Fine-grained sentiment analysis NoReC$_{fine}$ is a dataset[16] comprising a subset of the Norwegian Review Corpus (NoReC; Velldal et al., 2018) annotated for sentiment holders, targets, expressions, and polarity, as well as the relationships between them (Øvrelid et al., 2020). We here cast the problem as a graph prediction task and train a graph parser (Dozat and Manning, 2018; Kurtz et al., 2020) to predict sentiment graphs. The parser creates token-level representations which is the concatenation of a word embedding, POS tag embedding, lemma embedding, and character embedding created by a character-based LSTM. We further augment these representations with contextualised embeddings from each model. Models are trained for 100 epochs, keeping the best model on development F_1. For span extraction (holders, targets, expressions), we evaluate token-level F_1, and the common Targeted F_1 metric, which requires correctly extracting a target (strict) and its polarity. We also evaluate Labelled and Unlabelled F_1, which correspond to Labelled and Unlabelled Attachment in dependency parsing. Finally, we evaluate on Sentiment Graph F_1 (**SF**$_1$) and Non-polar Sentiment Graph F_1 (**NSF**$_1$. SF_1 requires predicting all elements (holder, target, expression, polarity) and their relationships (NSF_1 removes the polarity). A true positive is defined as an exact match at graph-level, weighting the overlap in predicted and gold spans for each element, averaged across all three spans. For precision we weight the number of correctly predicted tokens divided by the total number of predicted tokens (for recall, we divide instead by the number of gold tokens). We allow for empty holders and targets.

Sentence-level binary sentiment classification We further evaluate on the task of sentence-level binary (positive or negative) polarity classification, using labels that we derive from NoReC$_{fine}$ described above. We create the dataset for this by aggregating the fine-grained annotations to the sentence-level, removing sentences with mixed or no sentiment. The resulting dataset, NoReC$_{sentence}$, is made publicly available.[17] For the BERT models, we use the `[CLS]` embedding of the last layer as a representation for the sentence and pass this to a softmax layer for classification. We fine-tune the models in the same way as for

[14] https://github.com/ltgoslo/norne

[15] https://github.com/davidsbatista/NER-Evaluation

[16] https://github.com/ltgoslo/norec_fine

[17] https://github.com/ltgoslo/norec_sentence

	POS		
Model	BM	NN	Time
Stanza (Qi et al., 2020)	98.3	97.9	–
NorELMo$_{30}$	98.1	97.4	8
NorELMo$_{100}$	98.0	97.4	8
mBERT	98.0	97.9	245
NB-BERT	**98.7**	**98.3**	244
NorBERT	98.5	98.0	238

Table 2: Evaluation scores of the NorLM models on the POS tagging of Bokmål (BM) and Nynorsk (NN) test sets in comparison with other large pre-trained models for Norwegian. Running times in minutes are given for Bokmål.

the POS tagging task, training the models for 20 epochs and keeping the model that performs best on the development data. For ELMo models, we used a BiLSTM with global max pooling, taking ELMo token embeddings from the top layer as an input. The evaluation metric is macro F_1.

Negation detection Finally, the NoReC$_{\textit{fine}}$ dataset has recently been annotated with negation cues and their corresponding in-sentence scopes (Mæhlum et al., 2021). The resulting dataset is dubbed NoReC$_{\textit{neg}}$.[18] We use the same graph-based modeling approach as described for fine-grained sentiment above. We evaluate on the same metrics as in the *SEM 2012 shared task (Morante and Blanco, 2012): cue-level F_1 (CUE), scope token F_1 over individual tokens (ST), and the combined full negation F_1 (FN).

7.2 Results

We present the results for the various benchmarking tasks below.

POS tagging As can be seen from Table 2, NorBERT outperforms mBERT on both tasks: on POS tagging for Bokmål by 5 percentage points and 1 percentage point for Nynorsk. NorBERT is almost on par with NB-BERT on POS tagging. NorELMo models are outperformed by NB-BERT and NorBERT, but are on par with mBERT in POS tagging. Note that their adaptation to the tasks (extracting token embeddings and learning a classifier) takes 30x less time than with the BERT models.

[18] https://github.com/ltgoslo/norec_neg

Model	Bokmål	Nynorsk	Time
NorELMo$_{30}$	79.9	75.6	2
NorELMo$_{100}$	81.3	75.1	2
mBERT	78.8	81.7	14
NB-BERT	**90.2**	**88.6**	11
NorBERT	85.5	82.8	9

Table 3: NER evaluation scores (micro F_1) of the NorLM models on the NorNE test set in comparison with other large pre-trained models for Norwegian. Running time is given in minutes for the Bokmål part (on 1 NVIDIA P100 GPU).

See Figure 2 for the examples of training dynamics of the Nynorsk model.

Named entity recognition Table 3 shows the performance on the NER task. NB-BERT is the best on both Bokmål and Nynorsk, closely followed by NorBERT. Unsurprisingly, mBERT falls behind all the models trained for Norwegian, when evaluated on Bokmål data. With Nynorsk, it manages to outperform NorELMo. Bokmål is presumably dominant in the training corpora of both. However, in the course of fine-tuning, mBERT seems to be able to adapt to the specifics of Nynorsk. Since our ELMo setup did not include the fine-tuning step, the NorELMo models' adaptation abilities were limited by what can be learned from contextualised token embeddings produced by a frozen model. Still, when used on the data more similar to the training corpus (Bokmål), ELMo achieves competitive results even without any fine-tuning.

In terms of computational efficiency, the adaptation of ELMo models to this task requires 6x less time than mBERT or NB-BERT and 4x less time than NorBERT. Note also that the NorBERT model takes less time to fine-tune than the NB-BERT model (although the number of epochs was exactly the same), because of a smaller vocabulary, and thus less parameters in the model. Again, in this case an NLP practitioner has a rich spectrum of tools to choose from, depending on whether speed or performance on the downstream task is prioritised.

Fine-grained sentiment analysis Table 4 shows that NorBERT outperforms mBERT on all metrics and NB-BERT on all but SF_1, although the differences between NorBERT and NB-BERT are gen-

	Spans			Targeted	Parsing Graph		Sent. Graph		
Model	Holder F_1	Target F_1	Exp. F_1	F_1	UF_1	LF_1	NSF_1	SF_1	Time
Extraction [1]	42.4	31.3	31.3	–	–	–	–	–	–
$NorELMo_{30}$	55.1	55.3	57.2	**37.9**	49.0	41.2	40.9	34.5	446
$NorELMo_{100}$	58.8	55.8	56.8	37.1	49.7	41.2	41.5	34.2	434
mBERT	57.1	55.2	56.3	34.8	48.7	38.3	40.5	31.7	444
NB-BERT	61.3	56.1	57.9	36.0	49.7	41.9	40.7	**34.8**	404
NorBERT	**63.0**	**56.4**	**58.1**	36.9	**50.5**	**42.2**	**41.0**	**34.8**	438

Table 4: Average score of NorLM models on fine-grained sentiment (5 runs with set random seeds). **Bold** denotes the best result on each metric. [1] Span extraction baseline from Øvrelid et al. (2020), which uses a BiLSTM CRF with pretrained fastText embeddings.

Model	F_1
$NorELMo_{30}$	75.0
$NorELMo_{100}$	75.0
mBERT	67.7
NB-BERT	**83.9**
NorBERT	77.1

Table 5: F_1 scores for the different LMs models on the binary sentiment classification test set.

Model	CUE	ST	FN	Time
$NorELMo_{30}$	91.7	80.6	63.8	428
$NorELMo_{100}$	92.2	81.3	65.5	407
mBERT	**92.8**	**84.0**	**65.9**	353
NB-BERT	92.4	83.1	63.5	342
NorBERT	92.1	83.6	65.5	426

Table 6: Results of our negation parser, augmenting the features with token representations from each language model. The results are averaged over 5 runs.

erally small.

On this task the NorELMo models generally outperform mBERT as well. However, unlike in the previous tasks, the running times here are similar for BERT and ELMo models, since no fine-tuning was applied (the same is true for negation detection). We furthermore compare with the previous best model (Øvrelid et al., 2020), a span extraction model which uses a single-layer Bidirectional LSTM with Conditional Random Field inference, and an embedding layer initialized with fastText vectors trained on the NoWaC corpus. All approaches using language models outperform the previous baseline by a large margin on the span extraction tasks.[19] NorBERT, in particular, achieves improvements of 20.6 percentage points on Holder F_1 (24.9 and 25.8 on Target and Exp. F_1, respectively).

Binary sentiment classification Table 5 shows that NorBERT outperforms mBERT by 9.4 percentage points on sentiment analysis. However, it seems that in binary sentiment classification the sheer amount of training data starts to show its benefits, and NB-BERT outperforms NorBERT by 6.8 points. NorELMo models outperform mBERT by 7.3 points.

Figure 2 shows the training dynamics of the models.

Negation detection From Table 6 we can see that mBERT gives the best overall results, followed by NorBERT and $NorELMo_{100}$. NB-BERT and $NorELMo_{30}$ perform worse than the others on Scope token F_1 (ST) and full negation F_1 (FN), while all models perform similarly at cue-level F_1 (CUE). We hypothesise that the structural similarity of negation across many of the pretraining languages gives mBERT an advantage, but it is still surprising that it outperforms NB-BERT and NorBERT.

8 Future plans

In the future, separate Bokmål and Nynorsk BERT models are planned, and we further expect to train and evaluate models with a higher number of epochs over the training corpus. While we plan to develop additional monolingual Norwegian models based on other contextualised LM architectures

[19] Øvrelid et al. (2020) only perform span extraction. Therefore, it is not possible to compare the other metrics.

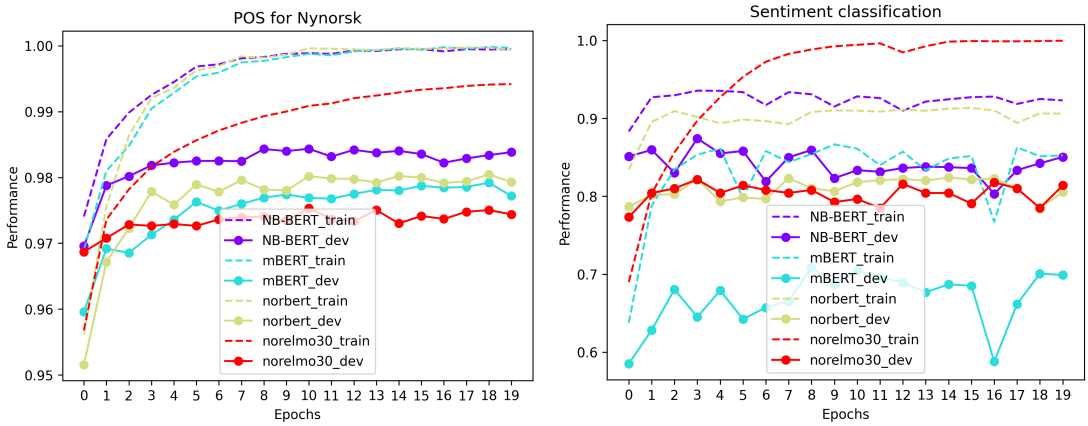

Figure 2: Per-epoch performance on training and development data for two of the tasks. Left: accuracy for POS tagging (Norwegian Nynorsk). Right: F_1 for binary sentiment classification.

beyond BERT and ELMo, we would also be interested to explore the usefulness of multilingual models restricted to Scandinavian languages. Further streamlining of the benchmarking process, in terms of both data access and computation of metrics, is something we also want to address in future work.

In addition, the ready availability of a highly optimised software stack on multiple HPC systems (published as part of NorLM) may contribute to other researchers developing very large contextualised language models for additional languages or language variants, e.g. domain- or application-specific sub-corpora. We hope that more pre-trained NLP models for Norwegian from both academy and industry will be openly released, making it possible to study the interplay between training corpora sizes, hyperparameters, pre-preprocessing decisions and performance in different tasks. At the same time, given the resource demands and sustainability issues related to training such models, we believe it will be important to coordinate efforts and we hope to collaborate closely with other players moving forward.

9 Summary

This paper has described the first outcomes of NorLM, an initiative coordinated by the Language Technology Group at the University of Oslo seeking to provide Norwegian (and Nordic) large-scale contextualised language models, while simultaneously focusing on maintaining a re-usable software environment for model development on national and Nordic HPC infrastructure. We have here described the training and testing of NorELMo and NorBERT – the first large-scale monolingual LMs for Norwegian. We have benchmarked the models across a wide array of Norwegian NLP tasks, also comparing to the multilingual mBERT model and another large-scale LM for Norwegian developed in parallel work, NB-BERT, trained on large amounts of text from historical sources. The results show that while the monolingual models tend to yield better results, which particular model ranks first varies across tasks. This underscores the importance of building an ecosystem of diversified models, accompanied by systematic benchmarking.

Acknowledgements

The NorLM resources are being developed on the Norwegian national super-computing services operated by UNINETT Sigma2, the National Infrastructure for High Performance Computing and Data Storage in Norway, as well as on the Finnish national supercomputing facilities operated by the CSC IT Center for Science. Software provisioning was financially supported through the European EOSC-Nordic project; data preparation and evaluation were supported by the SANT project (Sentiment Analysis for Norwegian Text), funded by the Research Council of Norway (grant number 270908). We are indebted to all funding agencies involved, the University of Oslo, and the Norwegian tax payer.

References

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing:

Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. 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.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2020. Parsbert: Transformer-based model for Persian language understanding. *arXiv preprint arXiv:2005.12515*.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

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 Edition of the Language Resources and Evaluation Conference*, Marseille, France, 2020.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Greek-Bert: The Greeks visiting Sesame street. In *11th Hellenic Conference on Artificial Intelligence*, pages 110–117.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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*.

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.

Andrey Kutuzov and Elizaveta Kuzmenko. 2017. Building web-interfaces for vector semantic models with the WebVectors toolkit. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 99–103, Valencia, Spain. Association for Computational Linguistics.

Petter Mæhlum, Jeremy Barnes, Robin Kurtz, Lilja Øvrelid, and Erik Velldal. 2021. Negation in Norwegian: an annotated dataset. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics*.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the National Library of Sweden – making a Swedish BERT. *arXiv preprint arXiv:2007.01658*.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 shared task: Resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 265–274, Montréal, Canada.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. Phobert: Pre-trained language models for vietnamese. *arXiv preprint arXiv:2003.00744*.

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, pages 1579–1585, Portorož, Slovenia.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.

Matej Ulčar and Marko Robnik-Šikonja. 2020. High quality ELMo embeddings for seven less-resourced languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4731–4738, Marseille, France.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian Review Corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 4186–4191, Miyazaki, Japan.

Erik Velldal, Lilja Øvrelid, and Petter Hohle. 2017. Joint UD parsing of Norwegian Bokmål and Nynorsk. In *Proceedings of the 21st Nordic Conference of Computational Linguistics*, pages 1–10, Gothenburg, Sweden.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. Bertje: A Dutch Bert model. *arXiv preprint arXiv:1912.09582*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.