Automatic Lemmatisation for Norwegian

Ahmet Yıldırım, Kristin Hagen, Dag Trygve Truslew Haug
Humit, Faculty of Humanities, University of Oslo
{ahmetyi,kristiha,daghaug}@uio.no

Abstract

We report on a new lemmatisation system for Norwegian, which is a particularly challenging language with two written standards, Bokmål and Nynorsk, that both have a lot of optionality. Our system covers both varieties and consists of a neural model that classifies words into rewrite rule classes that produce their lemma, as well as a large-scale computational lexicon of Norwegian that gives all possible inflections of a large part of the Norwegian vocabulary. We test different ways of combining these components. When evaluated with pure string-matching against the lemmas in the gold data, all systems perform approximately at the same level (99.1-99.2% on Bokmål and 98.5-98.6% on Nynorsk), but detailed error analysis shows that the computational lexicon reduces the number of true errors by more than half (reaching 99.6% accuracy on Bokmål and 99.3% on Nynorsk), as opposed to "surface errors" like using a different, but equally acceptable spelling variant of the correct lemma.

Keywords: lemmatisation, BERT models, computational lexicons, Norwegian

1. Introduction

Lemmatisation is the task of reducing variable forms of a word into a base form, typically the one used as the dictionary entry of the word. It is a task that has become somewhat less popular in recent years. It last figured in the SIGMORPHON shared tasks in 2019 (McCarthy et al., 2019). In later years, these tasks have focused on morphological generation, where lemmas are typically part of the input data rather than predicted. Within the natural language processing field, lemmatisation as a way of addressing the sparse data problem by eliminating morphological variants has largely given way to the use of embeddings. But for many text analysis tasks in the humanities, such as concept analysis or corpus-assisted lexicography, lemmatisation is clearly still very useful, precisely because it serves to group different word forms that are relevant to one and the same concept, or one and the same dictionary entry.

In this paper, we report on the development of a lemmatisation system for Norwegian. Norwegian is relatively well equipped with lexical resources, but nevertheless presents some unique challenges for lemmatisation related mainly to the fact that the language has two written varieties, Nynorsk and Bokmål, which are close enough to be sometimes indistinguishable and which both allow a considerable amount of spelling variation, including in lemma forms. Lemmatisation systems for Norwegian, including the one we present here, must cover both varieties and all the variation within them. Careful error analysis is therefore important to correctly evaluate the quality of such systems, to distinguish between substantial errors, where the system has picked the wrong lemma, or produced a nonexistent lemma, from those where the system has merely chosen a different way of spelling the correct lemma.

We test several different setups: a state-of-the-art system for neural multilingual lemmatisation, and two hybrid systems where the neural system intersects with a large-scale computational lexicon in two different ways. While the hybrid systems are language-specific and require resources that are not available for most languages, their performance in comparison with purely neural systems also tells us something about the strengths and weaknesses of the latter systems.

In Section 2 we detail some of the specific challenges in lemmatising Norwegian text. Then in Section 3 we survey related work in lemmatisation. Section 4 describes our different experiment setups and their components, while Section 5 evaluates their performance and provides detailed error analysis. Section 6 provides details on the implementation of the lemmatiser inside the overall Humit tagger and Section 7 concludes.

2. Lemmatisation for Norwegian

For historical reasons, there exist two different written standards for Norwegian, called *Bokmål* and *Nynorsk*. Bokmål is the majority standard, used by around 90% of the population, while Nynorsk is used by around 10% of the population, mainly in the western parts of the country (Karterud et al., 2024). The two varieties are close enough that sentences, especially short ones, may be indeterminate as to whether they are in Bokmål or Nynorsk. Many lemma forms are the same, but there are also lemmas that differ. Moreover, as language purism has been relatively strong within Nynorsk, there are a number of words, in particular of Low German origin, that are not considered proper in Nynorsk. Many of these words are rela-

Proceedings of the Workshop on Structured Linguistic Data and Evaluation, pages 93–103

tively common in the dialects and are, in fact, found in Nynorsk texts even if not officially acceptable. Furthermore, as the two varieties coexist within the same public discourse, it is relatively common to find e.g., quotes in Bokmål within an otherwise Nynorsk text and vice versa.

When lemmatising, we clearly want to use the standard that the text is written in, whether it is Bokmål or Nynorsk. But some practical questions arise: when we meet stray Bokmål words in Nynorsk text, we may want to deal with these words as proper Nynorsk words even if they officially do not exist. On the other hand, when we encounter longer stretches (often quotes) of Bokmål in Nynorsk text, we may want to deal with them in the same way as when we encounter English, German, or other foreign languages. This is, for example, what the Norwegian Dependency Treebank (Solberg et al. 2014; National Library of Norway 2014) does. In such cases, they apply the postag `unknown` and use the surface form as the lemma. This is the same policy as applied to e.g. English when it occurs in quotes and titles, but when it happens with Bokmål inside Nynorsk, it is confusing from a machine learning perspective that most of the words that are tagged in this way in fact also exist in Nynorsk. For example, in the sentence *Jeg har alltid lurt på hvem som vasker for vaskehjelpen.* ('I have always wondered who washes for the housekeeper'), there are four words that give this away as Bokmål; as a result, when this sentence is quoted inside a Nynorsk text, all words including the six neutral ones, receive the `unknown` tag and are lemmatised by their surface form, although in other parts of the corpus, where they occur in a Nynorsk sentence, they receive other tags and lemmas.

In addition to the complexity of having two written standards of Norwegian, both standards also allow for much more internal variation than what we typically find in other languages. Most of this variation is found in the morphology, but crucially, some of it affects the way the lemma is spelled. For example, infinitives in Nynorsk may end in *-e* or *-a*, and many verbs have an optional infixed *-j-*. This means that the infinitive 'to think' can be spelled *tenka, tenke, tenkja, tenkje*. As another example, the adverb 'forward' in Bokmål, which is very common as a prefix in Norwegian, can be spelled *fram* or *frem*. In such cases, it is not entirely clear what form we should target: one approach would be to use the form of the text we lemmatise, but that may not be clear, or the text may be inconsistent. Moreover, using the standard of the text that is being lemmatised makes it harder to do comparisons *across* texts. Finally, from a practical point of view, spelling variation of this type affects the evaluation of lemmatisation models, which is typically done by string matching.

As a final complication, we mention that Norwegian has undergone several spelling reforms in modern times, the most recent ones in 2005 (Bokmål) and 2012 (Nynorsk). For example, infinitives ending in *-a* were allowed for some but not all verbs in Bokmål until 2005. This affects lemmatisation and morphological analysis of historical texts.

3. Related Work

Automated lemmatisation of Norwegian text goes back at least to Fjeldvig and Golden (1984). This was an early attempt at so-called root lemmatisation, i.e. to link tokens to their root, ignoring, e.g. part of speech and derivational morphology, rather than to actual lemmas/dictionary entries. The system implemented rules that aimed to strip common prefixes and suffixes from Norwegian words, but without word-specific information as would be found in a lexicon. Subsequent work throughout the 1990s focused on developing a full computational lexicon of Norwegian, which eventually resulted in *Norsk ordbank* (Norwegian word bank, National Library of Norway 2022a,b). This resource is currently maintained by the Norwegian language council in collaboration with the University of Bergen. As mentioned above, there were several spelling reforms during this period. As a result, the paradigms in Norsk ordbank had to be marked with a `from_date` and a `to_date` attribute, indicating during which period a particular paradigm was valid. Handling this in a lemmatisation system is not entirely straightforward. On the one hand, one may want the system to be able to handle spellings that are no longer part of the norm. On the other hand, it is not desirable that the system produce forms that are not part of the norm as lemmas for forms that also has an alternative lemma that is part of the norm.

Norsk ordbank can be used to generate all possible inflected forms of the lemmas it contains, and their associated morphological features. Such a resource, in turn, can be used to generate all possible analyses (lemma + morphological features) of words in running text. However, it does not help with picking the correct analysis in context. To deal with this, a system of Constraint Grammar-based rules was developed called the Oslo-Bergen Tagger (Johannessen and Hagen, 2003). In later work (Johannessen et al., 2012), a statistical disambiguating system called OBT+stat was added to the rule-based system to deal with cases where the rule-based system accepted two or more alternative analyses. For lemmatisation, OBT+stat picked the globally most frequent lemma alternative in a large Norwegian web corpus (NoWaC, Guevara 2010). The reported accuracy for lemma-

tisation was 98.48% for Bokmål. The system did not handle Nynorsk.

Haug et al. (2023) showed that a neural architecture based on the NB-BERT model (Kummervold et al., 2021) on its own outperformed both the rule-based system for morphological tagging and a combined system using both the rules and a neural network. They suggested that rules may still be useful for lemmatisation, but left that for future research.

In the meantime, NorBench (Samuel et al., 2023) implemented the general UD lemmatisation pipeline as introduced by Straka et al. (2019) as part of a system to benchmark Norwegian language models. While the focus of that work is not to improve lemmatisation for Norwegian, but to assess language models' understanding of (among other things) Norwegian morphology, their experiments do provide useful benchmarks for purely neural lemmatisation of Norwegian. They report a 99.1% accuracy for lemmatisation on the Norwegian UD treebanks; however, they do not release the fine-tuned models. Our work tries to improve performance by combining this approach with the lexicon from *Norsk ordbank* by fine-tuning BERT models on the same dataset, the Norwegian Dependency Treebank.

Constructing hybrid systems for lemmatisation has been tried before, e.g. by Milintsevich and Sirts (2021), who started from a sequence-to-sequence model for lemmatisation (the Stanford neural lemmatisation system, Qi et al. 2020). However, we observe that on Danish and Swedish, which are closely related to Norwegian, this system generally performs worse than the system of Straka et al. (2019). We therefore chose to start from the latter.

4. Experimental Setup

The goal is to develop better lemmatisation capacities for Norwegian. To do this, we train a lemmatisation model by using the Norwegian Dependency Treebank dataset. Two of the configurations we test in this paper additionally use a morphological classifier as input for lemmatisation. Therefore, we also train a morphological classifier. We want to preserve the traditional tagset for Norwegian that has been used since the Oslo-Bergen Tagger in the nineties. We therefore target the tags of the original Norwegian Dependency Treebank rather than the converted UD ones (Øvrelid and Hohle 2016; National Library of Norway 2023), which means we train on the original treebank. The tagset covers both part-of-speech tags and a more detailed morphosyntactic description, including person, number, gender etc. Only the part-of-speech tag is used in the lemmatiser, but this lemmatiser is embedded in a larger system that also produces a full morphosyntactic description.

split	bokmål	nynorsk	total
train	280369	272355	552724
test	30650	30712	61362

Table 1: Data split word counts

Unlike the UD version, this treebank does not have an official train/test split, and so we create such a split with a 0.9/0.1 ratio, respectively.[1] Word counts for this split are given in Table 1. Note that this main train split is later subdivided into task-specific train and dev sets, whereas all evaluation is done on the main test set, to avoid data leaks between tasks, see below in Section 4.3.

Our previous experience with lemmatisation using neural models showed that even with the best models, which achieve very high overall accuracies, the occasional erroneous lemmas obtained can seem very strange for the human eye, for example, by violating hard phonotactic constraints in the language. Therefore, we experiment with different configurations of a single lemmatisation model, plus integration of a morphological classifier and a morphological lexicon. In the following, we give details of UDpipe lemmatisation pipeline, the lexicon, the machine learning setup, and the different experimental configurations.

4.1. The UDpipe Lemmatisation Pipeline

The idea of the UDpipe lemmatisation pipeline (Straka et al., 2019) is to find a rule that edits the word to get the lemma. In the first phase, the system looks at all word-lemma pairs and extracts rules, viz. edits that will rewrite the word as the lemma. Then, a model is trained that classifies input tokens into rule classes in a multi-class classification setting. In the inference phase, when a word and its rule are given, the UD lemmatisation pipeline applies the rule to the word to get the lemma.

The rules of the pipeline are created in a very general way to allow processing of as many languages as possible. A rule has four features: casing, prefix operations, suffix operations, and whether it represents a whole replacement with an absolute string lemma. The operations are copying/deletion/addition of characters in specific locations of the word with reference to the beginning (prefix) and end of the word (suffix). However, we

[1] This split is published in the `ndt_1.0_nob` and `ndt_1.0_nno` directories of https://github.com/humit-oslo/humit-tagger-sources, where all the training data for the Humit tagger system are available, including data for tasks not discussed in this article.

form	language	POS	lemma
arbeider	nno	verb	arbeide
arbeider	nob	noun	arbeid
arbeider	nob	noun	arbeide
arbeider	nob	noun	arbeider
arbeider	nob	verb	arbeide

Table 2: Sample lexicon entries. nno: Nynorsk, nob: Bokmål.

modified the code implemented by Samuel et al. (2023)[2], to make the rule extraction prefer suffix operations by inserting constant dummy characters at the beginning of both lemma and form. This was motivated by most inflections in Norwegian being suffixing, which also simplifies the classification tasks. To give an example for a rule, the word "planlagt" and the lemma rule "lower;-+e→-+ge" produce the lemma "planlegge". Here, the first part of the lemma rule indicates that the word will be lower-cased, and the second part indicates the following: add *e* at the end, replace the second last character with *g*, and replace the fourth last character with *e*.[3]

4.2. The Morphological Lexicon

The morphological lexicon we use is *Norsk ordbank*, mentioned above. From this lexicon, we extracted all possible forms, along with the information about language form (Bokmål, Nynorsk, or both), the lemma, and part of speech, yielding quadruples as in Table 2, which shows different forms from the root *arbeid* 'work'.

Norsk ordbank is a carefully hand-crafted resource that has been developed over a long period. This means that the analyses found there are, in principle, valid. Nevertheless, there are certain complications. First, we use part-of-speech to disambiguate; for example, in Table 2, if we know that the form *arbeider* is a verb, only the lemma *arbeide* is possible. However, the morphological analysis could be wrong. This can happen simply because the tagger makes an error, or because of discrepancies in the tags between the tagger and the lexicon. Such discrepancies arise because the part-of-speech classification that is used in the Norwegian Dependency Treebank differs in some details from the classification in *Norsk ordbank*. For example, the Norwegian Dependency Treebank treats words of location such as *her* 'here', *hjemme* 'at home' etc. as prepositions, while the lexicon uses the traditional classification of such words as adverbs. The differences are small, however, and we leave an eventual unification of the tagsets to future work.

Second, as we saw above, the lexicon contains word forms and inflections that are no longer considered correct. We include these to be able to deal with older texts and with non-standard language. This can sometimes lead to unwanted ambiguities. These could possibly be reduced by implementing a prioritization (only accept a non-standard analysis if no standard analysis exists), but we did not attempt such a strategy.

Third, Nynorsk often contains words that are considered Bokmål and which are therefore not listed as possible Nynorsk words. Here, we do implement a prioritization scheme: We enrich the vocabulary in Nynorsk with all Bokmål forms that do not otherwise occur in the lexicon. We do not similarly enrich the Bokmål lexicon with Nynorsk forms, because the amount of Nynorsk found in Bokmål texts is minuscule.

Fourth, many word forms do not occur in the lexicon at all. One reason is that compounding is a productive process in Norwegian. Recent words may also not yet be included. In these cases, the lexicon obviously does not give us a lemma. Similarly, in cases where the lexicon has more than one analysis, such as for *arbeider* as a noun in Table 2, the lexicon does not give us an analysis on its own.

4.3. Machine Learning Setup

To evaluate models during training, from the main training set, we create task-specific train and dev sets for individual tasks, lemmatisation, and morphological classification, each close to the size of the test set. In this way, while we keep the test set the same across these tasks, we create task-specific train and dev sets to balance class distributions and improve class coverage in the task-specific training processes.

The lemmatisation classifier is trained based on $\text{NorBert}_{3,\text{large}}$ since finetuning this model was reported to be the best performing model for this task (Samuel et al., 2023). Our modified rule extractor extracted 598 rules from the task-specific train set. We use the NorbertForTokenClassification module[4] to train this model. We treat a whole rule as one class, regardless of the rule components, such as casing, absolute rewriting of the lemma, or suffix.

Since the token classification task classifies tokens, the words with multiple tokens are handled as follows: The first token is assigned to the actual class, and the rest of the tokens to a default

[2]The repository for the code is: https://github.com/hplt-project/HPLT-WP4/blob/main/evaluation/ud/lemma_rule.py

[3]See Straka et al. (2019) for an explanation of the formalism.

[4]https://huggingface.co/ltg/norbert3-large/blob/main/modeling_norbert.py

class created for this purpose. This process also identifies the boundaries of words, which the tagger later uses to create its output. Therefore, the loss is computed for all tokens, but not specific tokens such as the beginning or the end token of the word.

Our initial experiments with the hyperparameters given by Samuel et al. (2023) performed poorly compared to the performance measure of 99.1% accuracy reported in their article. We believe this may be due to two reasons: 1) the report is based on a model trained for multiple UD tasks, having multiple classification heads, which leads the base model weights to be updated by training for multiple tasks, and 2) although the source of the datasets are the same, the training and test datasets in our experiments and the ones reported on in the article are different. Therefore, we experimented with several hyperparameters to achieve an accuracy comparable to the reported value, although the two models are not directly comparable.

Based on our experiences with the training process, we settled on using the following hyperparameters: batch size=8, number of epochs=50, and initial learning rate=0.00005. Other parameters are default parameters set by the HuggingFace Transformers Trainer library version 4.41. Throughout the training, we evaluate the model every 500 steps and select the best-performing model according to the accuracy score of the task-specific dev set. In other words, since we are looking for the best model, we update the weights in small batches with a low learning rate and evaluate the model at short intervals. Our lemmatisation model has a classification accuracy of 99.22% for the dev and 99.16% for the test set, which are close to the best previously reported (Samuel et al., 2023).

The morphological classifier model is trained with the same parameters for 267 morphological classes, where each class represents a set of morphological tags. We follow the process detailed in (Haug et al., 2023). We extract each morphological tag combination (such as [subst, prop]) from the training set, and consider each such combination as a different class that a token will be classified into. In line with recent work training and fine-tuning LLMs for Norwegian (Kummervold et al., 2021; Samuel et al., 2023), we train both the lemmatisation classifier and the morphological classifier on datasets consisting of both Bokmål and Nynorsk data. This is especially suitable because the base model that we are fine-tuning was also trained on both written varieties. In this training, the morphological classifier obtained by picking the best dev set classification accuracy has accuracy scores of 98.41% for the task-specific dev set and 98.27% for the test set. The upper bound of the trained model is given by the number of words in the test set that require labels not seen during training. This number is quite small: only 27 words, with one of these occurring twice for a total of 28 occurrences, i.e. less than 0.05%.

4.4. Different Experiment Configurations

We set up three different configurations to experiment with: `model_only`, `model_lexicon`, and `model_lexicon_disamb`.

`model_only` uses only the lemmatisation model. This configuration identifies the lemma rule class of each word and applies the rule to the word to get the lemma.

`model_lexicon` identifies the morphological class of words using the morphological classifier model. If the word is tagged `prop`, then it is considered a proper noun and the lemma is returned as the word itself, retaining upper case. If a proper noun is also tagged as `gen` and there is *s* at the end of the word, then it is considered in genitive form, and the lemma is produced by removing the last *s*. Otherwise, the `model_lexicon` checks the word in the morphological lexicon introduced in Section 4.2, and uses the lemma in this list. If a word (or a compound word) does not exist in the lexicon or if there are multiple suggestions from the lexicon, then the lemma is identified by applying the rule suggested by the lemmatisation model to the word, as in the `model_only` configuration.

`model_lexicon_disamb` applies the same steps as the `model_lexicon` configuration, but it tries to disambiguate among the lemmas if there are multiple suggestions for the word in the lexicon. This configuration, given a word, sorts the rule classes according to their probabilities retrieved from the lemmatisation model and selects the most probable one that produces the lemma among the suggestions of the morphological lexicon. If a word (or a compound word) does not exist in the lexicon, or if the lemmas obtained from the lemmatisation model and the lexicon do not share at least one lemma, then the first suggestion of the lemmatisation model is used, similar to the `model_only` configuration. We use the lemmatisation model's first suggestion to identify word boundaries for all configurations.

5. Evaluation and Discussion

This section evaluates and discusses the performance of the different configurations on the test set. Table 3 shows the number of errors and the accuracy rate of the three configurations on the Bokmål and Nynorsk data. We observe that Nynorsk is slightly harder than Bokmål, likely due to hav-

ing a more complex morphology. On the other hand, the differences between the various setups are small and look insignificant. However, as we will now see, the similarity in the number of errors hides large differences in the kind of errors that are made.

For the detailed error analysis, we focus on `model_only` and `model_lexicon_disamb`. Cursory investigation of the errors from `model_lexicon` showed that it is intermediate between `model_only` and `model_lexicon_disamb` in the types of errors that it makes.

Table 4 shows our categorization of the cases where there is a string mismatch between the gold lemma and the lemma produced by the setups `model_only` and `model_lexicon_disamb`. In the first group, we see real, indisputable errors. We have sorted these into the following categories: *Spelling errors* are errors caused by there being a spelling mistake in the source text. In the gold data these are lemmatised under the correctly spelled lemma. The lexicon obviously cannot handle such cases, as it does not contain misspellings. In principle, the trained model could be able to deal with them by finding a rule that rewrites them into their correct lemma. But since spelling errors are rare and unsystematic, we find small differences between `model_only` and `model_lexicon_disamb` in this category, at least for Bokmål. For Nynorsk, it does seem that the model is able to deal with some spelling mistakes. The second category is *wrong variety*, i.e. cases where the system produces a Bokmål lemma for Nynorsk text or vice versa. The trained model sometimes does this because it has been trained on Bokmål and Nynorsk jointly, as explained above. The third group is *capitalization*: these are due to the morphological tagger wrongly identifying a word as a proper noun (or vice versa). The fourth group is *abbreviations*: we see an interesting difference in how `model_lexicon_disamb` handles these worse in Bokmål. This is because the Bokmål test set contains abbreviations of political parties (*Ap.* for *Arbeiderpartiet*) spelled with a full stop; as these are proper nouns, `model_lexicon_disamb` uses the form as lemma, but the gold data in this case has the abbreviation without a stop. The fifth group is *other errors*. There are errors that cannot be further classified based only the surface forms, but require inspection into how the neural system and the rules interact. They will be discussed in more detail in Section 5.1. But we notice already that other errors are much less frequent with `model_lexicon_disamb` than with `model_only`.

However, there are many mismatches with the gold data which are arguably not real errors. The clearest cases are those where the gold data is simply wrong, which we call *gold errors*. As expected, these are about equally frequent in both configurations, although small differences arise if one configuration makes the same mistake as the gold data. The next group are *optional* forms: as explained in the introduction, many Norwegian lemmas have several equally acceptable spellings. The most common case here is actually the word *TV/tv* 'television' and its compounds, which can be spelled in capitals or in small letters. Finally, we come to the group which makes the biggest difference between `model_only` and `model_lexicon_disamb`, namely *adj/verb*. This is a classical problem in part-of-speech tagging for Norwegian (and other Germanic languages): When is a word form like *spist* 'eaten' an adjective and when is it a verb in the perfect participle form? This also translates into a lemmatisation problem because the morphological lexicon lemmatises the perfect participle as *spise* ('eat', the infinitive of the verb), but the adjective as *spist* ('eaten'). The Norwegian Dependency Treebank, which is the source of our training and test material, follows another standard by lemmatising also the adjective under the infinitive of the verb. For such cases, then, the morphological tagger has learned to categorize them as an adjective, but when this part of speech is used to look up the morphological lemma, we do not find the verbal lemma, only the adjective, and so `model_lexicon_disamb` makes a "mistake". However, this is merely a different convention.

The final group in Table 4 is *"Foreign" text*. Most of these are cases where a longer stretch of Bokmål is quoted in otherwise Nynorsk text, as discussed in Section 2. Swedish sometimes also causes problems because there are many shared words.[5] It would be beneficial if a standard could be established on how to handle such "code-switching" within Norwegian, as it does not seem correct to treat them as completely foreign words. We leave this for future work.

In sum, gold errors, optional variants, and adjective/verb errors are not true errors. However, they make up 54.7% and 53.2% of the errors made by `model_lexicon_disamb` in Bokmål and Nynorsk, respectively. For `model_only`, they only make up 28.1% and 21.3% of the errors. In other words, while the number of errors remains relatively constant between `model_only` and `model_lexicon_disamb`, the number of

[5] English, German, and other languages also occur in NDT as part of quotes or titles, but in this case, the forms are generally not existing Norwegian words, and so the model lemmatises them using the form as the lemma.

	Bokmål		Nynorsk	
Configuration	**errors**	**accuracy**	**errors**	**accuracy**
`model_only`	285	99.1%	456	98.5%
`model_lexicon`	240	99.2%	437	98.6%
`model_lexicon_disamb`	236	99.2%	447	98.5%

Table 3: Number of errors and accuracies on the test set

	Bokmål		Nynorsk	
Error type	**`model_only`**	**`model_lexicon_disamb`**	**`model_only`**	**`model_lexicon_disamb`**
Spelling error	19	22	13	27
Wrong variety	5	1	37	2
Capitalization	19	17	60	37
Abbreviation	3	15	2	2
Other errors	155	44	185	86
Gold errors	29	36	32	38
Optional	20	31	26	48
Adj/verb	31	67	39	152
“Foreign” text	4	3	62	62

Table 4: Error types

true errors is significantly reduced. This is shown in Table 5, which we believe gives a better measure of the performance improvement of `model_lexicon_disamb` over `model_only` than what we see in Table 3.

5.1. Further analysis of “other errors”

Table 4 already contains some analysis of the true errors that the systems make, based on directly observable properties of the test set or the produced forms. For Bokmål these are relatively constant across `model_only` and `model_lexicon_disamb`. For Nynorsk we see that using the lexicon improves capitalization errors and the production of forms of the wrong variety (i.e. Bokmål), although it also hurts the treatment of spelling errors. Nevertheless, for both varieties, the largest reduction comes in the category “other errors” in Table 4. We therefore proceed to a closer analysis of these errors.

Notice that it is hard to say what is going on in this category in the `model_only` configuration when there are no detectable properties of the surface forms like spelling errors or capitalization. The system simply picked the wrong lemma rewriting rule, and the usual difficulties in interpreting the BERT model makes it hard to say why. We therefore focus on the `model_lexicon_disamb` configuration.

Table 6 breaks down the “other errors” of Table 4 depending on whether they are due to errors in the lexicon or to classification errors by the neural model, which may arise whenever a POS-tagging error prevents successful lookup in the lexicon, or because there is no entry in the lexicon, or because there are several entries in the lexicon and the model needs to choose.[6]

We see that the true errors classified as “other errors” largely stem from the trained model. The morphological lexicon is a hand-crafted resource that has been maintained over a long period and does not contain many errors. However, our decision to include forms from the lexicon that are no longer acceptable sometimes induces an error whenever the lemma itself (and not just inflected forms) has had a variant spelling. For example, the Bokmål adjective *grønn* ‘green’ used to have a variant spelling *grøn* in its base form. As mentioned above, we include such forms to be able to deal with non-standard and historical texts, but this means that the neuter form, which has always only been *grønt*, can be lemmatised *grøn* or *grønn*, where only the latter is correct today.

Such cases are rare, and most errors are due to the neural model. A large part of these is due to part-of-speech errors. Such tagging mismatches may lead the system to dismiss a lexicon entry that it should have used. It should be noted that not all such tagging mismatches are true errors; as mentioned above, some are merely differences in the part-of-speech analysis between the lexicon and the annotation in the Norewgian Dependency Treebank.

When we look at the errors that the model makes, many of them are understandable. A common example is the weak/strong inflection ambiguity. Norwegian masculine nouns mark definiteness with a suffix *-en* as in e.g. *gutten* ‘the boy’ or

[6] Two errors that were due to an error in the code are not included here. The error has been corrected in the published implementation.

	Bokmål		Nynorsk	
Configuration	**errors**	**accuracy**	**errors**	**accuracy**
`model_only`	205	99.3%	359	98.8%
`model_lexicon_disamb`	102	99.6%	216	99.3%

Table 5: Errors on the test set, discounting gold errors, optional variants and adj/verb errors

Error type	**Bokmål**	**Nynorsk**
lexicon error	7	4
POS error	18	36
no lexicon entry	10	16
several lexicon entries	7	30

Table 6: Source of "other" errors in `model_lexicon_disamb`

hanen 'the rooster'. In some of these nouns (the so-called strong class), the lemma ends in a consonant: *gutt*. In others (the so-called weak class), the lemma ends in *-e*: *hane*. There are also ambiguous nouns: *listen* may be the definite form of either *liste* ('list') or *list* ('trim'). Similar facts hold for neuters with *-et* in the definite form. It is therefore perhaps not surprising that the model makes errors in adding a spurious *-e* to the lemma, or leaving out a required one. Ideally, the model would be able to generalise over common suffixes, but this is not always the case. For example, *beskyttelse* 'protection' contains the common deverbal suffix *-else*, while no Norwegian noun ends in *-els*. Still, `model_only` lemmatises *beskyttelsen* as *beskyttels*.

We should also point out that `model_only` sometimes makes serious mistakes that violate Norwegian phonotactic rules in such a way as to undermine a human's confidence in the lemmatisation. This is due to the rewrite rules not taking into account what letter is being changed, only the position where something changed. To use an English example, the rule that takes the word *women* to its lemma *woman* can be paraphrased as "change the last but one letter to *a*", and not as "change an *e* in the second last position to an *e*". When such a rule is applied wrongly, it can, depending on the phonological environment, yield very strange results. For example, there are quite a number of Norwegian irregular verbs that display a vowel alternation where an *a* in the past tense corresponds to an *e* in the infinitive lemma, e.g. *hang–henge* 'hang', *hjalp–hjelpe* 'help', *rakk–rekke* 'reach' and many more. From this, the system extracts the rule that the third letter from the end changes to an *e*, and then an additional *e* is inserted at the end. But when the system applies this rule to the verb form *strøk* 'stroked', the result is *steøke*, which contains an impossible vowel sequence *-eø-* that stands out in human eyes.

The overall number of errors is quite small, however, and it is useful to look at the cases where the lexicon does not at all provide an analysis to get an impression of how the model performs. These can be completely new words that have not yet been added to the lexicon, or words formed via compounding, which is a very productive process in Norwegian.

In total, there are 1785 occurrences of words in the Bokmål testset (of total 30650 words) that are not in the lexicon. For 1169 of these (65.5%), typically proper nouns and invariable words, the word form is the same as the lemma. For the Nynorsk test set (total 30712 words), there are 1872 that are not in the lexicon. 1063 of these (56.8%) have a lemma that is identical with the word form. In other words, defaulting to using the form as lemma whenever the word is not in the lexicon would yield a baseline performance of 57-66% on out-of-vocabulary items. However, as expected, the neural system performs much better than that, with an accuracy of 97.4% on these items in Bokmål and 95.8% in Nynorsk. This is, however, noticeably worse than the performance of `model_only` on the whole dataset, indicating that the words that are not in the lexicon are also harder for the neural system, likely because they are rare or nonexistent in the training data.

6. Implementation

We have incorporated the lemmatiser into our Norwegian morphological tagger, which is published in our repository [7]. This tagger prioritizes speed and ease of use, offering more features than the specific experiments described in this paper. While the reported experiments evaluate lexicon integration via the two distinct $\text{NorBert}_{3,\text{large}}$-based lemmatisation-rule and morphological classifier models, this implementation also handles sentence boundary and written standard (Nynorsk or Bokmål) detection.

Using a large model (323M parameters) for all of the mentioned tasks could become computationally intensive when processing large files. Therefore, during the implementation of the tagger, we have used a single base model and fine-tuned it with four classification heads for the

[7] https://huggingface.co/Humit-Oslo/humit-tagger-xs/tree/main

Input size	xs (cpu)	xs	small	base	large
1MB	00:56	00:55	00:59	01:10	02:00
10MB	06:13	04:32	04:50	06:18	13:57
50MB	30:01	19:16	20:50	27:55	58:34
100MB	58:35	39:13	41:35	57:42	2:11:51

Table 7: Execution durations of different sizes of the trained models on various sizes of input. Format: h:mm:ss

four different tasks mentioned here, including morphological and lemma-rule classification. Following our findings in this paper, we have implemented the lexicon in the tagger using the `model_lexicon_disamb` configuration. We have used various-sized NorBERT models (number of parameters from 15M to 323 M) as the base, enabling the tagger to run on a range of devices, including those with limited resources, such as in CPU-only mode with low RAM.

Table 7 gives an overview of the time required to run the tagger with varying input sizes and base models. When input is provided to the tagger, all classification heads are computed, although we use the tagger in a setting where only the lemmatisation and morphological tagging outputs are used. We tested the smallest model on a 14-core Intel Core i5 Linux-operating-system machine with 64 GB RAM, and we tested all models on an NVIDIA GeForce RTX 2080 GPU with 12 GB RAM. While our implementation tags a 1MB plain text file ranging from 56 seconds to 2 minutes, depending on the setting, Stanza (Qi et al., 2020), a Python NLP package based on neural networks, POS-tags and lemmatises the same text in 6 minutes on the same CPU and 3 minutes on the same GPU environment. This makes our implementation competitive in terms of speed with one of the widely-utilized lemmatisation and POS-tagger implementations. For usability, we have implemented the tagger as downloadable models in our HuggingFace repository[8], which requires only a few lines of code to run.

7. Conclusion and Future Work

Despite accuracy scores that are already in the high nineties, there is still scope for improving neural lemmatisation systems by combining them with high-quality lexical resources. Lemmatisation is an inherently difficult task for LLM-based systems because the output set, the vocabulary of the target language, is open-ended. The UD lemmatisation system of Straka et al. (2019) partly overcomes this by classifying into rewrite rules from form to lemma, rather than directly to lemmas, but the number of classes remain large, and performance is improved when handcrafted lexicons are used instead of the model whenever possible.

As we have seen above, this strategy noticeably reduces the number of true errors in the Norwegian data. However, when accuracy rates are as high as above 99%, it becomes very important to do careful error analysis. For example, around 10% of the errors are due to mistakes in the gold data. And many more errors are due to cases where multiple analyses would be acceptable.

This points to a general difficulty in integrating neural, data-driven systems with hand-crafted lexicons, namely that the training data for the neural system may rely on different annotation conventions from those of the lexical resources. As *Norsk ordbank* is a resource that is continually maintained by the Language Council of Norway, it is desirable to make it easy to integrate new words from this source. By contrast, the Norwegian Dependency Treebank is no longer actively maintained, which means that it will be a one-time effort to change its annotation practices to match those of *Norsk ordbank* (as long as this resource does not change its annotation practices).

Another line of work we will consider in the future is to increase the number of cases where the lexicon can be used by implementing compound analysis. As we have mentioned, compounding is a productive process in Norwegian and in many, probably most, cases, both compound members are in the lexicon.

Finally, with performance at such high levels, it will clearly be useful to construct new and more challenging datasets. This would include historical data, non-standard data with more frequent spelling mistakes, specialized language with rare terminology, etc.

8. Limitations

This work only deals with Norwegian. As such, the techniques and the configurations we use are only applicable for a high-resource language that has both enough text data to train a BERT model (Vaswani et al., 2017), enough annotated data to fine-tune this model for lemmatisation and morphological classification, *and* rich lexical resources such as a morphological lexicon. Very few languages have all of this. However, there is a lesson for planning the development of new resources for low-resource languages: rich, structured lexical information can still make an impact. As pointed out by one of the reviewers, there are languages with relatively few resources that still have extensive dictionaries. Many dictionaries will

[8] https://huggingface.co/Humit-Oslo/humit-tagger-xs/tree/main

not give all possible realisations of lemmas, as *Norsk ordbank* does, but even a simple dictionary can be used to filter the output of a machine learning system and thereby avoid e.g. phonotactically impossible lemmas that risk undermining human confidence in the output.

9. Bibliographical References

Tove Fjeldvig and Anne Golden. 1984. Automatisk rotlemmatisering (automatic root lemmatization) [in Norwegian]. In *Proceedings of the 4th Nordic Conference of Computational Linguistics (NODALIDA 1983)*, pages 84–103, Uppsala, Sweden. Centrum för datorlingvistik, Uppsala University, Sweden.

Emiliano Raul Guevara. 2010. NoWaC: a large web-based corpus for Norwegian. In *Proceedings of the NAACL HLT 2010 Sixth Web as Corpus Workshop*, pages 1–7, NAACL-HLT, Los Angeles. Association for Computational Linguistics.

Dag Haug, Ahmet Yildirim, Kristin Hagen, and Anders Nøklestad. 2023. Rules and neural nets for morphological tagging of Norwegian - results and challenges. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 425–435, Tórshavn, Faroe Islands. University of Tartu Library.

Janne Bondi Johannessen and Kristin Hagen. 2003. Parsing nordic languages (panola) norsk versjon. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2002*, pages 89–95. Museum Tusculanums Press.

Janne Bondi Johannessen, Kristin Hagen, André Lynum, and Anders Nøklestad. 2012. Obt+stat: A combined rule-based and statistical tagger. In Gisle Andersen, editor, *Exploring Newspaper Language*, pages 51–66. John Benjamins Publishing Company.

Thomas Karterud, Kristin Rogge Pran, and Mathilde Horvei. 2024. Språkrådets brukerog befolkningsundersøkelse. https://sprakradet.no/wp-content/uploads/Sprakradets-Bruker-og-befolkningsundersokelse-2024.pdf. Accessed: 2026-02-16.

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.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Kirill Milintsevich and Kairit Sirts. 2021. Enhancing sequence-to-sequence neural lemmatization with external resources. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3112–3122, Online. Association for Computational Linguistics.

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).

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.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Palatkina. 2023. NorBench – a benchmark for Norwegian language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 618–633, Tórshavn, Faroe Islands. University of Tartu Library.

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 (LREC'14)*, pages 789–795, Reykjavik, Iceland. European Language Resources Association (ELRA).

Milan Straka, Jana Straková, and Jan Hajic. 2019. UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morpho-

logical categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

10. Language Resource References

National Library of Norway. 2014. *The Norwegian dependency treebank*. National Library of Norway. PID https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-10/.

National Library of Norway. 2022a. *Norsk ordbank bokmål*. National Library of Norway. PID https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-5/.

National Library of Norway. 2022b. *Norsk ordbank nynorsk*. National Library of Norway. PID https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-41/.

National Library of Norway. 2023. *Norwegian UD treebank*. National Library of Norway. PID https://www.nb.no/sprakbanken/ressurskatalog/oai-nb-no-sbr-83/.