Joint UD Parsing of Norwegian Bokmål and Nynorsk

Erik Velldal and **Lilja Øvrelid** and **Petter Hohle**
University of Oslo
Department of Informatics
{erikve,liljao,pettehoh}@ifi.uio.no

Abstract

This paper investigates interactions in parser performance for the two official standards for written Norwegian: Bokmål and Nynorsk. We demonstrate that while applying models across standards yields poor performance, combining the training data for both standards yields better results than previously achieved for each of them in isolation. This has immediate practical value for processing Norwegian, as it means that a single parsing pipeline is sufficient to cover both varieties, with no loss in accuracy. Based on the Norwegian Universal Dependencies treebank we present results for multiple taggers and parsers, experimenting with different ways of varying the training data given to the learners, including the use of machine translation.

1 Introduction

There are two official written standards of the Norwegian language; Bokmål (literally 'book tongue') and Nynorsk (literally 'new Norwegian'). While Bokmål is the main variety, roughly 15% of the Norwegian population uses Nynorsk. However, language legislation specifies that minimally 25% of the written public service information should be in Nynorsk. The same minimum ratio applies to the programming of the Norwegian Public Broadcasting Corporation (NRK).

The two varieties are so closely related that they may in practice be regarded as 'written dialects'. However, lexically there can be relatively large differences. Figure 1 shows an example sentence in both Bokmål and Nynorsk. While the word order is identical and many of the words are clearly related, we see that only 2 out of 9 word forms are identical. When quantifying the degree of lexical overlap with respect to the treebank data we will be using, (Section 3) we find that out of the 6741 non-punctuation word forms in the Nynorsk development set, 4152, or 61.6%, of these are unknown when measured against the Bokmål training set. For comparison, the corresponding proportion of unknown word forms in the Bokmål development set is 36.3%. These lexical differences are largely caused by differences in productive inflectional forms, as well as highly frequent functional words like pronouns and determiners.

In this paper we demonstrate that Bokmål and Nynorsk are different enough that parsers trained on data for a given standard alone can not be applied to the other standard without a vast drop in accuracy. At the same time, we demonstrate that they are similar enough that mixing the training data for both standards yields better performance. This also reduces the complexity required for parsing Norwegian, in that a single pipeline is enough. When processing mixed texts (as is typically the case in any real-world setting), the alternatives are to either (a) maintain two distinct pipelines and select the right one by applying an initial step of language identification (for each document, say), or (b) use a single-standard pipeline only and accept a substantial loss in accuracy (on the order of 20–25 percentage points in LAS and 15 points in tagger accuracy) whenever text of the non-matched standard is encountered.

In addition to simply combining the labeled training data as is, we also assess the feasibility of applying machine translation to increase the amount of available data for each variety. All final models and data sets used in this paper are made available online.[1]

2 Previous Work

Cross-lingual parsing has previously been proposed both for closely related source-target lan-

[1] https://github.com/erikve/bm-nn-parsing

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 1–10,
Gothenburg, Sweden, 23-24 May 2017.

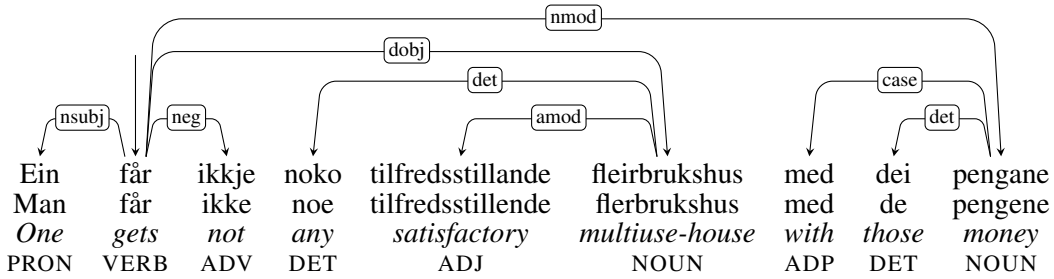

Figure 1: Example sentence in Nynorsk (top row) and Bokmål (second row) with corresponding English gloss, UD PoS and dependency analysis.

guage pairs and less related languages. This task has been approached via so-called 'annotation projection', where parallel data is used to induce structure from source to target language (Hwa et al., 2005; Spreyer et al., 2010; Agić et al., 2016) and as delexicalized model transfer (Zeman and Resnik, 2008; Søgaard, 2011; Täckström et al., 2012). The basic procedure in the latter work has relied on a simple conversion procedure to map part-of-speech tags of the source and target languages into a common tagset and subsequent training of a delexicalized parser on (a possibly filtered version of) the source treebank. Zeman and Resnik (2008) applied this approach to the highly related language pair of Swedish and Danish, and Skjærholt and Øvrelid (2012) extended the language inventory to also include Norwegian, and showed that parser lexicalization actually improved parsing results between these languages.

The release of universal representations for PoS tags (Petrov et al., 2012) and dependency syntax (Nivre et al., 2016) has enabled research in cross-lingual parsing that does not require a language-specific conversion procedure. Tiedemann et al. (2014) utilize statistical MT for treebank translation in order to train cross-lingual parsers for a range of language pairs. Ammar et al. (2016) employ a combination of cross-lingual word clusters and embeddings, language-specific features and typological information in a neural network architecture where one and the same parser is used to parse many languages.

In this work the focus is on cross-standard, rather than cross-lingual, parsing. The two standards of Norwegian can be viewed as two highly related languages, which share quite a few lexical items, hence we assume that parser lexicalization will be beneficial. Like Tiedemann et al. (2014), we experiment with machine translation of training data, albeit using a rule-based MT system with no word alignments. Our main goal is to arrive at the best joint model that may be applied to both Norwegian standards.

3 The Norwegian UD Treebank

Universal Dependencies (UD) (de Marneffe et al., 2014; Nivre, 2015) is a community-driven effort to create cross-linguistically consistent syntactic annotation. Our experiments are based on the Universal Dependency conversion (Øvrelid and Hohle, 2016) of the Norwegian Dependency Treebank (NDT) (Solberg et al., 2014).

NDT contains manually annotated syntactic and morphological information for both varieties of Norwegian; 311,000 tokens of Bokmål and 303,000 tokens of Nynorsk. The treebanked material mostly comprises newspaper text, but also includes government reports, parliament transcripts and blog excerpts. The UD version of NDT has until now been limited to the Bokmål sections of the treebank. For the purpose of the current work, the Nynorsk section has also been automatically converted to Universal Dependencies, making use of the conversion software described in Øvrelid and Hohle (2016) with minor modifications.[2]

UD conversion of NDT Nynorsk Figure 1 provides the UD graph for our Nynorsk example sentence. The NDT and UD schemes differ in terms of both PoS tagset and morphological features, as well as structural analyses. The conversion therefore requires non-trivial transformations of the dependency trees, in addition to mappings of tags and labels that make reference to a combination

[2]The data used for these experiments follows the UD v1.4 guidelines, but its first release as a UD treebank will be in v2.0. For replicability we therefore make our data available from the companion Git repository.

of various kinds of linguistic information. For instance, in terms of PoS tags, the UD scheme offers a dedicated tag for proper nouns (PROPN), where NDT contains information about noun type among its morphological features. UD further distinguishes auxiliary verbs (AUX) from main verbs (VERB). This distinction is not explicitly made in NDT, hence the conversion procedure makes use of the syntactic context of a verb; verbs that have a non-finite dependent are marked as auxiliaries.

Among the main tenets of UD is the primacy of content-words. This means that content words, as opposed to function words, are syntactic heads wherever possible, e.g., choosing main verbs as heads, instead of auxiliary verbs and promoting prepositional complements to head status instead of the preposition (which is annotated as a case marker, see Figure 1). The NDT annotation scheme, on the other hand, largely favors functional heads and in this respect differs structurally from the UD scheme in a number of important ways. The structural conversion is implemented as a cascade of rules that employ a small set of graph operations that reverse, reattach, delete and add arcs, followed by a relation conversion procedure over the modified graph structures (Øvrelid and Hohle, 2016). It involves the conversion of verbal groups, copula constructions, prepositions and their complements, predicative constructions and coordination, as well as the introduction of specialized dependency labels for passive arguments, particles and relative clauses.

Since the annotation found in the Bokmål and Nynorsk sections of NDT follow the same set of guidelines, the conversion requires only minor modifications of the conversion code described in Øvrelid and Hohle (2016). These modifications target (a) a small set of morphological features that have differing naming conventions, e.g., *ent* vs *eint* for singular number, and *be* vs *bu* for definiteness, and (b) rules that make reference to closed class lemmas, such as quantificational pronouns and possessive pronouns.

4 Experimental Setup

This section briefly outlines some key components of our experimental setup. We will be reporting results of two pipelines for tagging and parsing – one based on TnT and Mate and one based on UDPipe – described in the following.

TnT & Mate The widely used TnT tagger (Brants, 2000), implementing a 2nd order Markov model, achieves high accuracy as well as very high speed. TnT was used by Petrov et al. (2012) when evaluating the proposed universal tag set. Solberg et al. (2014) found the Mate dependency parser (Bohnet, 2010) to have the best performance for parsing of NDT, and recent dependency parser comparisons (Choi et al., 2015) have also found Mate to perform very well for English. The fast training time of Mate also facilitates rapid experimentation. Mate implements the second-order maximum spanning tree dependency parsing algorithm of Carreras (2007) with the passive-aggressive perceptron algorithm of Crammer et al. (2006) implemented with a hash kernel for faster processing times (Bohnet, 2010).

UDPipe UDPipe (Straka et al., 2016) provides an open-source C++ implementation of an entire end-to-end pipeline for dependency parsing. All components are trainable and default settings are provided based on tuning towards the UD treebanks. The two components of UDPipe used in our experiments comprise the MorphoDiTa tagger (Straková et al., 2014) and the Parsito parser (Straka et al., 2015).

MorphoDiTa implements an averaged perceptron algorithm (Collins, 2002) while Parsito is a greedy transition-based parser based on the neural network classifier described by Chen and Manning (2014). When training the components, we use the same parametrization as reported in Straka et al. (2016) after tuning the parser for version 1.2 of the Bokmål UD data. For the parser, this includes form embeddings of dimension 50, PoS tag, FEATS and arc label embeddings of dimension 20, and a 200-node hidden layer. For each experiment, we pre-train the form embeddings on the training data (i.e., the raw text of whatever portion of the labeled training data is used for a given experiment) using word2vec (Mikolov et al., 2013), again with the same parameters as reported by Straka et al. (2015) for a skipgram model with a window of ten context words.

Parser training on predicted tags All parsers evaluated in this paper are both tested *and* trained using PoS tags predicted by a tagger rather than gold tags. Training on predicted tags makes the training set-up correspond more closely to a realistic test setting and makes it possible for the parser

to adapt to errors made by the tagger. While this is often achieved using jackknifing (*n*-fold training and tagging of the labeled training data), we here simply apply the taggers to the very same data they have been trained on, reflecting the 'training error' of the taggers. We have found that training on such 'silver-standard' tags improves parsing scores substantially compared to training on gold tags (Hohle et al., 2017). In fact, Straka et al. (2016) also found that this set-up actually yields higher parsing scores compared to 10-fold tagging of the training data. Of course, the test sets for which we evaluate the performance is still unseen data for the taggers.

Data split For Bokmål we use the same split for training, development and testing as defined for NDT by Hohle et al. (2017). As no pre-defined split was established for Nynorsk we defined this ourselves, following the same 80-10-10 proportions and also taking care to preserve contiguous texts in the various sections while also keeping them balanced in terms of genre.

Evaluation The taggers are evaluated in terms of tagging accuracy (*Acc* in the following tables) while the parsers are evaluated by labeled and unlabeled attachment score (*LAS* and *UAS*). For the TnT tagger, accuracy is computed with the `tnt-diff` script of the TnT-distribution, and scores are computed over the base PoS tags, disregarding morphological features. Mate is evaluated using the MaltEval tool (Nilsson and Nivre, 2008). For the second pipeline, we rely on UDPipe's built-in evaluation support, which also implements MaltEval.

5 Initial experiments

5.1 'Cross-standard' parsing

This section presents the initial results of tagging and parsing the two written standards for Norwegian – Bokmål (BM) and Nynorsk (NN). Table 1 shows the results for both TnT+Mate and UDPipe. In both cases, we also show the effect of 'cross-standard' training and testing, i.e., training models on the Bokmål data and testing them on the Nynorsk data, and *vice versa*.

Across all metrics and data configurations, we see that UDPipe performs slightly better than TnT+Mate, but in particular with respect to tagger accuracy. However, a direct comparison of the scores is not really meaningful, for several reasons. First, the UDPipe components make use of more of the information available in the training data than TnT+Mate. For example, the tagger uses information about lemmas, while both the tagger and parser use morphological features. In addition, UDPipe is trained with the development set as validation data, selecting models from the iterations with the best performance.

	Train	Test	Acc	LAS	UAS
TnT+Mate	BM	BM	96.67	84.13	87.34
		NN	81.02	59.96	67.26
	NN	NN	95.81	82.09	85.39
		BM	79.73	59.85	66.02
UDPipe	BM	BM	97.55	84.16	87.07
		NN	83.06	61.11	68.61
	NN	NN	97.11	82.63	85.56
		BM	82.17	62.04	68.67

Table 1: Results on the UD development data for tagging and parsing the two written standards for Norwegian, Bokmål (BM) and Nynorsk (NN), including 'cross-standard' training and testing.

More interestingly, for both pipelines we see that performance suffers dramatically when a model trained for one variety is applied to the other. This means that one can not assume (as is sometimes done, often by necessity due to unavailable resources) that tools created for, say, Bokmål can be applied to Nynorsk without a substantial increase in errors.

5.2 The effect of data size

To gauge the effect that the size of the training set has on the performance of taggers and parsers applied to the Norwegian UD treebank, we computed learning curves where models are trained on partitions that are created by successively halving the training set (selecting every *n*th sentence). With data set size shown on a logarithmic scale, Figure 2 plots both tagger accuracy (left) and parser LAS (right) – where Mate and the UDPipe parser (Parsito) are applied to the tags predicted by TnT and the UDPipe tagger (MorphoDiTa) respectively. Note that the word embeddings used by Parsito are pre-trained on the corresponding subset of training data for each run.

A couple of interesting things can immediately be gleaned from these results: We see that while the TnT+Mate pipeline seems to be doing better than UDPipe when training on the smaller partitions, UDPipe outperforms TnT+Mate when train-

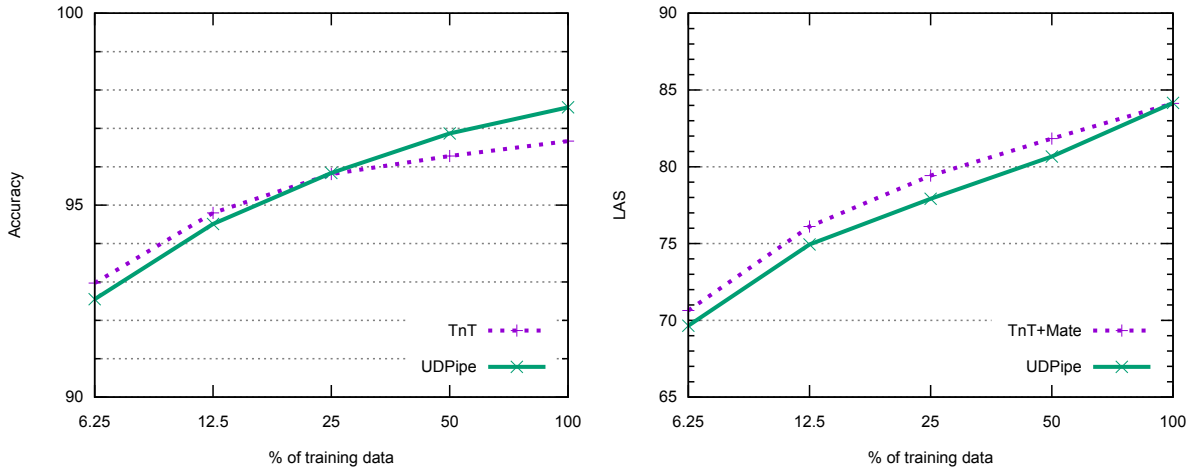

Figure 2: Learning curves when training the two pipelines TnT+Mate and UDPipe on successively halved partitions of the Norwegian Bokmål training set (using a log scale), while testing on the development set. UPoS tagging accuracy to the left; labeled attachment score to the right.

ing on the full training set. Moreover, in all cases, we observe a roughly log-linear trend where improvements are close to being constant for each n-fold increase of data. The trends also seem to indicate that having access to even more labeled data could improve performance further.

5.3 Motivating the further experiments

The 'cross-standard' experiments in Section 5.1 showed that models trained on labeled data for one of the two varieties of written Norwegian perform poorly when applied to the other. For all tested configurations, we observe a loss of between 20 and 25 percentage points in labeled attachment score compared to training and testing on one and the same variety. At the same time, it is important to realize that the results for 'within-standard' processing of either the Bokmål or Nynorsk treebank data in isolation, correspond to an idealized setting that is not representative of how written Norwegian is encountered 'in the wild'. In the news sources, blogs, government reports and parliament transcripts that form the basis for the treebank, both varieties of Norwegian will occur, intermixed. In practice, this means that the actual parsing results can be expected to lie somewhere in between the extremes reported in Table 1. Of course, a language identification module could be trained and applied as a pre-processing step for selecting the appropriate model, but in practice it would be much more convenient if we were able to have a single model that could process both varieties equally well.

In the next section, we look into various ways of mixing training data for the two written standards of Norwegian in order to create improved models for cross-standard joint processing. Moreover, given the empirical indications in Section 5.2 that more labeled training data could benefit the taggers and parsers, this strategy is also motivated by wanting to improve the absolute results for each standard in isolation.

6 Joint models

In this section we test the effects of combining the training data for Bokmål and Nynorsk, as well as extending it through machine translation.

6.1 Mixed training data

In a first round of experiments we simply concatenate the training sections for Bokmål and Nynorsk. The results can be seen in the row 'BM+NN' in Table 2. For both pipelines and both language varieties we observe the same trend: Despite a loss in tagging accuracy, parsing performance improves when compared to training on just a single variety (rows 'BM' or 'NN'). While effectively doubling the size of the training data, we do not see the same factor of improvement as for the learning curves in Figure 2, but we nonetheless see an increase in LAS of up to one additional percentage point. It is important to note that the results for 'BM+NN' represents using *joint* tagging and parsing pipelines across both written standards: For each set-up (TnT+Mate and UDPipe) we train a single pipeline, and then apply

	Bokmål				Nynorsk			
	Train	**Acc**	**LAS**	**UAS**	**Train**	**Acc**	**LAS**	**UAS**
Mate	BM	**96.67**	84.13	87.34	NN	**95.81**	82.09	85.39
	BM+NN	96.29	84.97	88.04	BM+NN	95.18	83.13	86.22
	BM+MT	96.32	**85.45**	**88.47**	NN+MT	94.98	**83.63**	**86.82**
	BM+NN+MT	96.30	85.05	88.12	BM+NN+MT	94.97	83.47	86.65
UDPipe	BM	**97.55**	84.16	87.07	NN	**97.11**	82.63	85.56
	BM+NN	97.01	84.65	87.42	BM+NN	96.43	82.81	85.84
	BM+MT	97.17	85.03	87.97	NN+MT	96.16	82.47	85.57
	BM+NN+MT	96.83	**85.10**	**88.01**	BM+NN+MT	96.15	**83.20**	**86.28**

Table 2: Development results for Bokmål and Nynorsk tagged and parsed with TnT+Mate and UDPipe, training on Bokmål or Nynorsk alone (rows BM or NN), mixed (BM+NN), or each combined with machine-translated data (BM+MT or NN+MT), or everything combined, i.e., the original and translated versions of both the Bokmål and Nynorsk training data (BM+NN+MT).

the same pipeline to both the Nynorsk and Bokmål development sets.

As a control experiment, to better understand to what extent the improvements are due only to larger training sets or also to the use of mixed data, we ran the same experiments after down-sampling the combined training set to the same size as the originals (simply discarding every second sentence). For TnT+Mate and UDPipe respectively, this gave a LAS of 82.81 and 82.77 for Bokmål, and 81.47 and 80.86 for Nynorsk. We see that while training joint models on the down-sampled mixed data gives slightly lower results than when using the full concatenation (or using dedicated single-standard models), it still provides a robust alternative for processing mixed data, given the dramatically lower results we observed for cross-standard testing in Section 5.1.

6.2 Machine-translated training data

The results above show that combining training data across standards can improve parsing performance. As mentioned in the introduction, though, there is a large degree of lexical divergence between the two standards. In our next suite of experiments, we therefore attempt to further improve the results by automatically machine-translating the training texts. Given the strong degree of structural equivalence between Norwegian Bokmål and Nynorsk, we can expect MT to yield relatively accurate translations. For this, we use the two-way Bokmål–Nynorsk MT system of Unhammer and Trosterud (2009), a rule-based shallow-transfer system built on the open-source MT platform Apertium (Forcada et al., 2011).

The raw text passed to Apertium is extracted from the full-form column of the UD CoNLL training data (translating the lemmas does not give adequate results). The only sanity-checking we perform on the result is ensuring that the number of tokens in the target translation matches that of the source. In cases where the token counts diverge – for example when the Bokmål form *fortsette* ('continue') is translated to Nynorsk as *halde fram* ('keep on') – the sentence is left in its original source form. For the NN→BM translation, this is the case for almost 4% of the sentences. The direction BM→NN appears to be slightly harder, where almost 13% of the sentences are left untranslated.

We tested the translated training data in two ways: 1) Training single-standard pipelines, for example training on the original Bokmål data and the Nynorsk data translated to Bokmål, and 2) training on all the available training data combined, i.e., both of the original versions and both of the translated versions, in effect increasing the amount of training data by a factor of four.

The results for the development data are shown in Table 2. Adding the MT data reinforces the trend observed for mixing the original training sets: Despite that PoS tagging accuracy typically (though not always) decreases when adding data, parsing accuracy improves. For the TnT+Mate pipeline, we see that the best parser performance is obtained with the single-standard models including the MT data, while UDPipe achieves the best results when using the maximal amount of training data. Coupled with the parser learning curves in Figure 2, this observation is in line with the expectation that neural network architectures both require and benefit more from larger training samples, but recall the caveat noted in Section 5.1

about how the scores are not directly comparable. Finally, note that this latter configuration, i.e., combining both of the original training sets with both of the translated versions, again corresponds to having a single joint model for both Bokmål and Nynorsk. Also for TnT+Mate, we see that this configuration yields better results than our previous joint model without the MT data.

6.3 Caveat on morphology

Although the development results demonstrate that the various ways of combining the training data lead to increased parser performance, we saw that the tagging accuracy was slightly reduced. However, the UDPipe tagging component, MorphoDiTa, performs additional morphological analysis beyond assigning UPoS tags. It also performs lemmatization and assigns morphological features, and in particular for the first of these tasks the drop in performance for the joint models is more pronounced. For example, when comparing the Bokmål development results for the UDPipe model trained on the original Bokmål data alone versus Bokmål and Nynorsk combined, the lemmatization accuracy drops from 97.29% to 95.18% (and the morphological feature accuracy drops from 96.03% to 95.39%). This is not surprising. Given the close similarities of Bokmål and Nynorsk, several words in the two variants will have identical inflected forms but different lemmas, introducing a lot of additional ambiguity for the lemmatizer. The drop in lemma accuracy is mostly due to a handful of high-frequent words having this effect, for example the verb forms *var* ('was') or *er* ('is') which should be lemmatized as *være* in Bokmål and *vere* in Nynorsk. However, for the taggers trained on the maximal training data where we include the machine-translated versions of both varieties, the lemma accuracy really plummets, dropping to 86.19% (and morphological feature accuracy dropping to 93.79%). Again, this is as expected, given that only the full-forms of training data were translated.

In our parsing pipeline, lemmas are not used and so this drop in accuracy does not affect downstream performance. However, for applications where lemmatization plays an important role, a joint tagger should either be trained without the use of the MT data (or an initial single-standard lemmatizer should be used to lemmatize this data after translation), and ideally should be made to take more context into consideration to be able to make more accurate predictions.

7 Held-out results

For the held-out results, we focus on testing the two joint models, i.e., (1) estimating models from the original training sets for Nynorsk and Bokmål combined, as well as (2) augmenting this further with the their MT versions (translating each variety into the other). We contrast the performance of these joint models with the results from training on either of the original single-standard training sets in isolation, including cross-standard testing. Table 3 summarizes the results for both pipelines – TnT+Mate and UDPipe – for the held-out sections of the treebanks for both of the Norwegian written varieties – Bokmål (BM) and Nynorsk (NN).

In terms of relative performance, the outcome is the same as for the development data: The joint models give better parsing performance across all configurations, compared to the dedicated single-standard models, despite reduced tagger accuracy. In terms of absolute figures, we see that UDPipe has the best performance.

It is also interesting to note that the UDPipe parser appears to be more robust to the noise introduced with MT data, and that this may even have had the effect of mitigating overfitting: While we observe a slight drop in performance for the single-variety models when moving from development to held-out results, the effect is the opposite for the joint model trained on the MT data. This effect is most pronounced for the Nynorsk data, which is also known to have the most translation errors in the training data.

Finally, note that while our parser scores are stronger than those previously reported for UDPipe on Norwegian (Bokmål only) (Straka et al., 2016), there are several reasons why the results are not directly comparable. First, we here use version 1.4 of the UD treebank as opposed to version 1.2 for the results of Straka et al. (2016), and secondly, the embeddings generated by word2vec are non-deterministic, meaning that strictly speaking, different UDPipe models for the same training data can only be directly compared if reusing the same embeddings.

8 Future work

Immediate follow-up work will include using a larger unlabeled corpus for pre-training the word

		BM			NN		
	Training	**Acc**	**LAS**	**UAS**	**Acc**	**LAS**	**UAS**
Mate	BM	**96.31**	83.80	87.04	81.66	60.51	67.55
	NN	80.32	60.64	67.13	**95.55**	81.51	85.06
	BM+NN	95.98	84.74	87.83	95.06	83.11	86.42
	BM+NN+MT	95.79	**84.88**	**87.89**	94.78	**83.87**	**87.16**
UDPipe	BM	**97.07**	83.42	86.28	83.35	60.95	68.15
	NN	82.92	62.85	69.66	**96.80**	82.40	85.38
	BM+NN	96.49	84.20	86.90	96.27	83.46	86.24
	BM+NN+MT	96.48	**85.31**	**88.04**	96.05	**84.17**	**87.18**

Table 3: Held-out test results for Norwegian Bokmål and Nynorsk tagged and parsed with TnT+Mate and UDPipe, using either the Bokmål or Nynorsk training data alone (rows BM or NN), Bokmål and Nynorsk mixed (BM+NN), or Bokmål and Nynorsk combined with machine-translated data, i.e., the original versions of both varieties as well as the translations of each into the other (BM+NN+MT).

embeddings used by UDPipe's Parsito parser. For this, we will use the Norwegian Newspaper Corpus which consists of texts collected from a range of major Norwegian news sources for the years 1998–2014, and importantly comprising both the Bokmål and the Nynorsk variety. Another direction for optimizing the performance of the pipelines is to use different training data for the different components. This is perhaps most important for the UDPipe model. While the parser benefits from including the machine-translated data in training, the tagger performs better when using the combination of the original training data. This is mostly noticeable when considering not just the accuracy of the UPoS tags but also the morphological features, which are also used by the parser. Finally, while the experimental results in this paper are based on the UD conversion of the Norwegian Dependency Treebank, there is of course no reason to expect that the effects will be different on the original NDT data. We plan to also replicate the experiments for NDT, and make available both pre-trained joint and single-standard models for this data set as well.

9 Conclusion

This paper has tackled the problem of creating a single pipeline for dependency parsing that gives accurate results across both of the official varieties for written Norwegian language – Bokmål and Nynorsk. Although the two varieties are very closely related and have few syntactic differences, they can be very different lexically. To the best of our knowledge, this is the first study to attempt to build a uniform tool-chain for both language standards, and also to quantify cross-standard performance of Norwegian NLP tools in the first place.

The basis of our experiments is the Norwegian Dependency Treebank, converted to Universal Dependencies. For Bokmål, this treebank conversion was already in place (Øvrelid and Hohle, 2016), while for the Nynorsk data, the conversion has been done as part of the current study. To make our results more robust, we have evaluated and compared pipelines created with two distinct set of tools, each based on different learning schemes; one based on the TnT tagger and the Mate parser, and one based on UDPipe.

To date, the common practice has been to build dedicated models for a single language variant only. Quantifying the performance of models trained on labeled data for a single variety (e.g., the majority variety Bokmål) when applied to data from the other (Nynorsk), we found that parsing accuracy dramatically degrades, with LAS dropping by 20–25 percentage points. At the same time, we found that when combining the training data for both varieties, parsing performance in fact increases for both. Importantly, this also eliminates the issue of cross-standard performance, as only a single model is used. Finally, we have shown that the joint parsers can be improved even further by also including machine-translated versions of the training data for each variety.

In terms of relative differences, the trends for all observed results are consistent across both of our tool chains, TnT+Mate and UDPipe, although we find the latter to have the best absolute performance. Our results have immediate practical value for processing Norwegian, as it means that a single parsing pipeline is sufficient to cover both official written standards, with no loss in accuracy.

References

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Alonso Martínez, Natalie Schluter, and Anders Søgaard. 2016. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. One parser, many languages. *arXiv preprint arXiv:1602.01595*.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Thorsten Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference*, Seattle, WA, USA.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning*, pages 957–961, Prague, Czech Republic.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 740–750, Doha, Qatar.

Jinho D. Choi, Joel Tetreault, and Amanda Stent. 2015. It Depends: Dependency Parser Comparison Using A Web-Based Evaluation Tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 387–396, Beijing, China.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, PA, USA.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singe. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies. A cross-linguistic typology. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 4585–4592, Reykjavik, Iceland.

Mikel L. Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Petter Hohle, Lilja Øvrelid, and Erik Velldal. 2017. Optimizing a PoS tagset for Norwegian dependency parsing. In *Proceedings of the 21st Nordic Conference of Computational Linguistics*, Gothenburg, Sweden.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11(3).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jens Nilsson and Joakim Nivre. 2008. MaltEval: An evaluation and visualization tool for dependency parsing. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 161–166, Marrakech, Morocco.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the International Conference on Language Resources and Evaluation*, Portorož, Slovenia.

Joakim Nivre. 2015. Towards a Universal Grammar for Natural Language Processing. In *Computational Linguistics and Intelligent Text Processing*, volume 9041 of *Lecture Notes in Computer Science*, pages 3–16. Springer International Publishing.

Lilja Øvrelid and Petter Hohle. 2016. Universal Dependencies for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, Slovenia.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A Universal Part-of-Speech Tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 2089–2096, Istanbul, Turkey.

Arne Skjærholt and Lilja Øvrelid. 2012. Impact of treebank characteristics on cross-lingual parser adaptation. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 187–198, Lisbon, Portugal.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 682—-686, Portland, Oregon.

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*, pages 789–795, Reykjavik, Iceland.

Kathrin Spreyer, Lilja Øvrelid, and Jonas Kuhn. 2010. Training parsers on partial trees: A cross-language comparison. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Milan Straka, Jan Hajič, Jana Straková, and Jan Hajič jr. 2015. Parsing universal dependency treebanks using neural networks and search-based oracle. In *Proceedings of Fourteenth International Workshop on Treebanks and Linguistic Theories*, Warsaw, Poland.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, Portorož, Slovenia.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics*, Montreal, Canada.

Jörg Tiedemann, Željko AgićZeljko, and Joakim Nivre. 2014. Treebank translation for cross-lingual parser induction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 130–140.

Kevin Brubeck Unhammer and Trond Trosterud. 2009. Reuse of Free Resources in Machine Translation between Nynorsk and Bokmål. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 35–42, Alicante.

Dan Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, Hyderabad, India.