Neural and Linear Pipeline Approaches to Cross-lingual Morphological Analysis

Çağrı Çöltekin
University of Tübingen
Department of Linguistics
ccoltekin@sfs.uni-tuebingen.de

Jeremy Barnes
University of Oslo
Department of Informatics
jeremycb@ifi.uio.no

Abstract

This paper describes Tübingen-Oslo team's participation in the cross-lingual morphological analysis task in the VarDial 2019 evaluation campaign. We participated in the shared task with a standard neural network model. Our model achieved analysis F1-scores of 31.48 and 23.67 on test languages Karachay-Balkar (Turkic) and Sardinian (Romance) respectively. The scores are comparable to the scores obtained by the other participants in both language families, and the analysis score on the Romance data set was also the best result obtained in the shared task. Besides describing the system used in our shared task participation, we describe another, simpler, model based on linear classifiers, and present further analyses using both models. Our analyses, besides revealing some of the difficult cases, also confirm that the usefulness of a source language in this task is highly correlated with the similarity of source and target languages.

1 Introduction

Morphological analysis is one of the basic tasks in natural language processing (NLP). The need for morphological analysis becomes particularly important in processing morphologically rich languages, where analysis of words can both be challenging and fruitful. Morphological analysis can be useful in downstream NLP tasks as well as being useful for (linguistic) research.

Traditionally, morphological analyzers have been developed using finite state transducers (FSTs). Finite-state morphological analyzers define a lexicon and a set of rules to specify both morphotactics and morpho-phonological (or orthographic) alternations. The resulting rule-based system is compiled into a finite state transducer which is capable of analyzing a given word to an underlying linguistic representation. The resulting FSTs are fast, and can be used for a range of tasks from stemming/lemmatization to full morphological analysis. As well as transducing word forms to a linguistic analysis, they can also be used in reverse to generate the word form(s) of a given linguistic representation.

Finite-state morphological analyzers have been used successfully for a broad range of NLP tasks, and are available for most of the world's major languages. Finite-state analyzers also exist for all of the languages that are featured in this shared task (examples of such analyzers include, Tzoukermann and Liberman, 1990; Altintas and Cicekli, 2001; Armentano-Oller et al., 2006; Çöltekin, 2010; Kessikbayeva and Cicekli, 2014; Washington et al., 2014; Forcada et al., 2011; Tyers et al., 2010). On the downside, developing these analyzers requires substantial expert effort,[1] which in some cases may not even exist, e.g., for languages with few speakers where experts are also hard to find. A potential solution to aid developing morphological analysis tools is to use unsupervised methods. Earlier attempts to develop unsupervised morphological analysis tools, mostly within Morpho Challenge shared tasks (Kurimo et al., 2010), returned rather mixed, often sub-optimal results (see Hammarström and Borin, 2011, for a survey).

Another approach for obtaining morphological analyses for languages without a morphological analyzer is based on transfer learning, which has become a widespread approach in NLP and related disciplines rather recently (Yarowsky et al., 2001; Faruqui and Kumar, 2015; Johnson et al., 2017; Barnes et al., 2018). The general idea is to train a supervised machine learning model that predicts analyses of word forms in a target language using gold-standard analyses that exist in other related languages.

[1] Access to an analyzer for a closely-related language may reduce the development time and effort considerably (Washington et al., 2014).

The present shared task, cross-lingual morphological analysis, takes the second approach. Track 1 of the task that we participated in aims to analyze words in a 'surprise' language, given gold-standard analyses of words in languages in the same language family. The second track included some additional resources (see Zampieri et al. (2019) for further details about the task).

The present task is also strongly related to the series of SIGMORPHON (re)inflection tasks (Cotterell et al., 2017, 2018), where the emphasis is in generation of the inflected forms rather than producing an analysis. Another difference between the present task and the inflection tasks is also the level of ambiguity. In inflection tasks, especially in context, ambiguity level is rather low, making it less pressing to produce multiple results, while dealing with ambiguity is more important in morphological analysis.

We developed multiple systems for the task. Our main system was a neural encoder–decoder architecture, where we used a recurrent network as encoder and lemma decoder, but unlike many earlier examples, we do not consider POS tags and morphological features as part of the output sequence. Although they share the encoder, the tags are predicted by multi-layer feed-forward neural classifiers. The second, simpler method is a set of linear SVM classifiers. Besides describing both models, we report further experiments and analyses, including a comparison of the models, a detailed error analysis, and a set of experiments investigating the roles of individual source languages in transfer learning.

2 Models

2.1 Linear baseline

Recently, the dominating approaches to morphology learning tasks have been neural models, particularly recurrent neural networks. However, linear models provide surprisingly good performance in some tasks (e.g., Çöltekin and Rama, 2016, 2018), with the added advantage that they are computationally cheaper to train and tune, and often exhibit less variance than modern neural architectures. Although our submissions were recurrent encoder/decoder architectures, we also implemented a fully linear approach to solve the task.

Our linear model is a pipeline model with components for predicting lemma, POS, and morphological features separately. After having experimented with different orders, our final pipeline first predicts the lemma, then POS tags, and finally the morphological features. In all parts, we use (multi-class) linear SVM models.

Lemma prediction is a two-step process, using two separate classifiers. The first classifier predicts the stem, the prefix shared by both the word form and the lemma. Subsequently, the second classifier predicts the possibly null string to be added to the lemma. For example, for the word *uçağını* 'his/her/their airplane-ACC' (Crimean Tatar), whose lemma is *uçaq*, the first classifier segments the word form as *uça·ğını*, and the second classifier predicts the string '*q*' to be appended to the stem. The features for both classifiers are the overlapping character n-grams, before and after the segmentation point.

POS tag prediction is also based on a classifier with character n-gram features. The n-grams for the (predicted) lemma and the suffixes after the segmentation point are used as features for a multi-class linear classifier.

Morphological tag prediction is similar to the POS tag prediction. In the linear model reported here, we treat the whole feature string as class labels. We have also experimented with multiple classifiers per feature, and a standard multi-label approach predicting individual `Feature=Value` pairs. However, in our preliminary experiments the monolithic single classifier yielded better performance on the development sets. In addition, it also offers an easier way to obtain n-best predictions during decoding.

Decoding follows the above order for the complete analysis of a given word form. At each step, we use a threshold value to pick n-best results. All predictions with a distance from the decision boundary larger than the threshold is produced, and passed to the next predictor in the pipeline.

2.2 Recurrent encoder/decoder

Our neural model follows a similar pipeline approach, again, predicting lemma, POS tag and morphological features one by one. The overall architecture is presented in Figure 1. The order of components are different from the linear model.[2] Another notable difference from the linear model

[2] The choice is due to computational convenience. We did not investigate the effects of the order of components on the overall prediction performance.

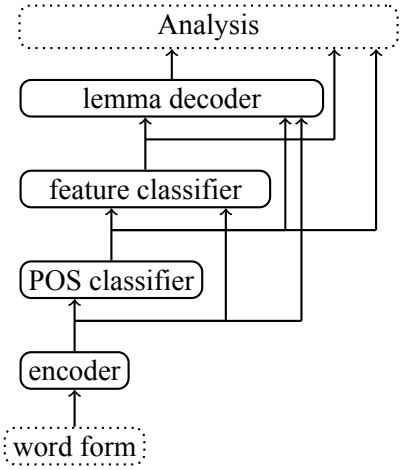

Figure 1: Overall architecture of the neural model.

is that the neural model shares some components during training, where components of the linear model are all trained/tuned individually.

The encoder is a bidirectional recurrent networks with gated-recurrent units (GRU, Cho et al., 2014) operating on input characters. Characters are passed through an embedding layer before being fed to the recurrent encoder. In this study, the embeddings are trained within the task, we do not use pre-trained character embeddings. We do not use intermediate representations of the input word either. Only the final representation, concatenation of forward and backward RNNs, is fed into the other parts of the network.

The POS classifier is a feed-forward component with two hidden layers with relu units followed by a softmax classifier.

The morphological feature classifier consists of multiple feed-forward networks for each morphological feature. Similar to the POS classifier we use two hidden layers with relu activation, followed by a softmax classifier for each morphological feature. The target values for each morphological feature are the feature values observed in the training data as well as a special 'not applicable' value. The morphological feature classifiers are trained jointly.

The lemma decoder is a recurrent decoder with GRU units. The initial symbol to the network is a special 'end of sequence symbol' and otherwise predictions of the previous time step are fed to the recurrent unit as input. The hidden state of the recurrent unit is initialized with the final output of the encoder. Similar to the encoder, the characters are embedded as continuous vectors before being fed into the recurrent layer. The embedding layers of the encoder and the lemma decoder are not shared. The output of the encoder, along with the tag predictions are fed to a softmax classifier at each step, which outputs the characters of the lemma.

We train the model in multiple steps. First the model is trained to guess POS tags, then morphological features, and finally the lemmas. While training a model further in the pipeline we initialize the encoder (and embedding) weights with the weights from the previous step, but freeze the weights of the classifier(s) of the previous step(s).

During decoding, we follow the same order. For POS tags we predict all POS tags until the total probability assigned by the softmax classifier exceeds a particular threshold. During the lemma prediction, we predict a lemma whenever probability of end-of-sequence symbol reaches to a defined threshold. We do not predict multiple values for the morphological features.

3 Experimental setup

3.1 Data and preprocessing

The CMA task included data from two language families, Romance (ROA) and Turkic (TRK). Since we participate only on track 1, we only make use of morphological analyses released by the shared task organizers. The reader is referred to Zampieri et al. (2019) for detailed description of the data set. We give a brief description of the data set here.

The Turkic (TRK) data consisted of training samples from Bashkir (bak), Kazakh (kaz), Kyrgyz (kir), Tatar (tat) and Turkish (tur), Turkic development data came from Crimean Tatar (crh), and test data was from Karachay-Balkar (krc). The Romance (ROA) data consisted of training samples from Catalan (cat), French (fra), Italian (ita), Portuguese (por) and Spanish (spa). Romance development and test data were from Asturian (ast) and Sardinian (srd) respectively. The number of word forms along with the number of lemmas, tags (POS and morphological feature combinations) and analyses per word form for each language is presented in Table 2.

For both language families, the task involves predicting possibly multiple analyses consisting of a lemma, a POS tag, and a set of morphological feature–value pairs for each word form (examples

word form	lemma	POS	morphological features
desgaste	*desgaste* 'wear'	NOUN	Gender=Masc|Number=Sing
	desgastar 'to wear (out)'	VERB	Mood=Sub|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin
karın	*kar* 'snow'	NOUN	Case=Gen
	kar 'snow'	NOUN	Case=Nom|Number[psor]=Sing|Person[psor]=2
	kar 'snow'	VERB	Mood=Imp|Number=Plur|Person=2|Valency=2|VerbForm=Fin
	karı 'wife'	NOUN	Case=Nom|Number[psor]=Sing|Person[psor]=2
	karın 'stomach'	NOUN	Case=Nom

Table 1: Examples taken from Spanish (ROA track) and Turkish (TRK track) training data for the morphological prediction task.

family	lcode	words	analysis	lemma	pos	tag
TRK	bak	9 999	1.46	1.10	1.08	1.38
	kaz	9 995	1.67	1.18	1.18	1.59
	kir	10 000	1.41	1.11	1.09	1.32
	tat	10 000	1.42	1.10	1.08	1.37
	tur	9 990	1.97	1.20	1.11	1.95
(dev)	crh	999	1.25	1.05	1.03	1.23
(test)	krc	8 768				
ROA	cat	10 000	1.44	1.15	1.28	1.43
	fra	9 986	1.67	1.15	1.29	1.66
	ita	9 998	1.55	1.21	1.35	1.54
	por	9 999	1.41	1.08	1.11	1.41
	spa	9 999	1.39	1.15	1.28	1.39
(dev)	ast	1 000	1.46	1.13	1.26	1.44
(test)	srd	9 998				

Table 2: Statistics on individual languages of CMA analysis data. The column 'words' is the number of word forms, the other columns indicate the ambiguity, e.g., 'pos' indicates number of part-of-speech tags per word form. 'analysis' indicate the full-analysis ambiguity, 'tag' indicates ambiguity of full morphological tag (combination of the POS and morphological features).

shown in Table 1). The POS tag set used for both languages consist of nouns, adjectives, adverbs, and verbs. The number of unique morphological feature–value combinations is 89 in the ROA training set, and 1 013 in the TRK training set.

3.1.1 Transliteration

The Turkic data set includes languages that use two different scripts. Turkish and Crimean Tatar uses the Latin script, while the other languages in this data set are written with the Cyrillic script. To our knowledge there are no standard way to transliterate between Turkic languages.[3] As a result, we used a rather ad hoc transliteration that tries to keep similarly-sounding letters of Cyrillic used in the languages of the training set, and the version of the Latin script used in Turkish and Crimean Tatar.

3.2 Evaluation

Following the official evaluation script, we report precision, recall and F1-scores, for lemmas, tags (combination of POS tags and morphological features) and full analysis (combination of all) for each word form. In some experiments we also report separate scores for POS tags and morphological features. We compare our models against the competition baseline, which is a neural machine translation model (Silfverberg and Tyers, 2019).

3.3 Linear model

All classifiers in our linear models are linear SVM classifiers. For multi-class classifiers (all except the stemmer), we use one-vs-rest multi-class strategy. All models were implemented in scikit-learn Python library (Pedregosa et al., 2011) using liblinear back end (Fan et al., 2008).

We tuned each classifier separately using random search on the development set, where all languages in the training set were used without any weighting scheme. Tuning involved classifier regularization parameter, maximum n-gram order used as features and threshold parameter for each classifier that affect the number of predictions produced during decoding. The resulting parameter values are listed in Table 3. The threshold of 0.00 in Table 3 indicates a single prediction, which means the configuration chosen by our tuning procedure produces only a single-best analysis on the Turkic data set, and producing multiple predictions only for the POS tags on the Romance data.

[3] Standard/documented transliteration methods from Cyrillic to Latin script exists for most languages. However, these methods are often developed better readability of the resulting text in English, which often diverges from the version of Latin script used in the Turkic languages.

Classifier	parameter	Romance	Turkic
POS	C	0.08	0.70
	threshold	−0.50	0.00
	n-grams	5	9
Lemma	C (seg)	0.02	0.02
	C (suffix)	3.70	0.52
	seg. threshold	0.00	0.00
	n-grams	5	7
Features	C	0.70	4.80
	threshold	0.00	0.00
	n-grams	10	5

Table 3: Hyperparameter for the linear model determined with a random search through the parameter space. A threshold value of 0.00 means only a single prediction. Values for n-grams are the maximum n-gram order used as features.

3.4 Neural model

For the neural model, we fixed the model architecture after initial experimentation. We used an embedding size of 64. Both forward and backward GRU layers in the encoder learned 512-dimensional representations, resulting in 1 024 hidden units in the lemma decoder. We used a dropout of 0.50 before the encoder (after embeddings) and before each classifier. We tuned the models using random search for optimum threshold values, selecting the model that resulted in the best overall analysis F1-score on the development set. The best scores were obtained for both language families with a POS tag threshold of 0.70 and a lemma threshold of 0.50. The neural model was implemented with Tensorflow (Abadi et al., 2015) using Keras API (Chollet et al., 2015).

4 Results and discussion

4.1 Performance on test and dev sets

Official evaluation results of submitted (neural) system in comparison to the shared-task baseline provided by the organizers are presented in Table 4. The system obtained good results on the ROA test set (Sardinian) in comparison to the baseline and the other participants. It predicted the tags particularly well, which also lead to the best analysis score despite lower lemma scores. The results on the TRK test set (Karachay-Balkar) are below shared-task baseline which was the clear winner on this language family by surpassing the scores of the other participants as well.

family/model	Analysis	Lemma	Tag
ROA			
NN	23.67	31.36	61.33
Baseline	22.94	31.56	51.88
TRK			
NN	31.53	52.74	38.93
Baseline	39.79	54.94	44.56

Table 4: Official results obtained by our neural model in comparison to the shared-task baseline. The scores are F1-scores.

The scores of our submitted model, the linear baseline described in Section 2.1, and the baseline results as reported by the organizers are presented in Table 5 with some additional detail. Since our models were tuned to perform well on the development set without exploiting the similarities or differences between the training and the test languages, it is not surprising that the test set results are substantially lower than the scores we obtained on the development set. However, the result on Table 5 also offers a few interesting observations.

Our NN model obtains better scores than the competition baseline on both language families. In contrast to the test set, on the development set the difference on the Turkic data is more pronounced. Our model yields an analysis F1-score approximately 16 percentage points (pp.) higher than the baseline on the dev set, while this difference is approximately 8 pp. in favor of the baseline on the test set. A likely reason for the difference is the tuning procedure. An untuned model is likely to be more general, and hence may do better on a surprise language. Another potential reason for the difference can be related to the transliteration process (see Section 4.2 for further discussion).

In comparison to the neural model, the linear model performs worse on the ROA data set. However, it performs competitively on the TRK data set, even yielding better lemma predictions than the neural model. The linear pipeline predicts the lemmas first, while neural model also makes use of the earlier POS and feature predictions during predicting lemmas. Although propagation of the error may affect the lemma predictions of the neural model adversely, it also has more information.

The difference in performance between linear and neural models across language families may also be due to their morphological typol-

family	Analysis			Lemma			Tag			POS			Morphology		
model	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
ROA															
NN	41.14	45.12	43.04	60.07	60.82	60.44	60.61	65.62	63.01	87.58	79.70	83.45	66.56	73.78	69.99
Linear	32.99	42.54	37.16	53.00	60.00	56.29	49.38	62.84	55.30	63.05	70.34	66.50	61.86	75.33	67.94
Baseline	42.57	43.33	42.94	55.91	60.60	58.16	60.13	59.50	59.81	–	–	–	–	–	–
TRK															
NN	47.84	53.36	50.45	72.33	73.59	72.95	55.70	62.07	58.71	87.80	81.67	84.63	55.91	62.88	59.19
Linear	43.86	53.95	48.38	78.53	82.38	80.41	47.80	58.76	52.72	82.15	84.28	83.20	47.84	58.76	52.74
Baseline	31.24	38.44	34.47	56.30	59.06	57.65	38.52	47.35	42.48	–	–	–	–	–	–

Table 5: Detailed results on the development set in comparison to the our linear baseline (Linear) as well as the competition baseline (Baseline). Besides the F1 scores (F) we also present precision and recall. Last two groups, 'POS' and 'Morphology' columns are a breakdown of the 'Tag' scores to part of speech tags and morphological features, respectively.

ogy. Predicting agglutinating morphology of Turkic languages with linear models may be easier, due to more transparent mappings between the morph(eme)s and relevant tags. On the other hand, the more fusional nature of Romance languages may require combining multiple pieces of information (possibly non-linearly) for successful predictions.

4.2 Effect of source language

In transfer learning, a natural question to ask is how useful a particular source language, or combination of source languages can be for a given target language. To test the effects of the source language in analyzing a target language, we used all individual languages in the training set as source, and tested on all training and development languages for both families. Due to computational convenience, we performed these experiments using only the linear model. The results of this 'cross-training' experiments are presented in Figure 2. The presented scores are the overall best analysis F1-scores obtained after a random search through the space of hyperparameters listed in Table 3. The diagonal presents the results of tests on the training languages, hence, only useful for an approximate upper bound achievable by the model on the given language.

An interesting observation from Figure 2 is that while analyzing the Romance development data (Asturian), the score obtained using only Spanish (40.79) is better than the results we obtained using the complete training set (37.16). In Turkic languages, no single language is better than the overall score we obtained. However, using only Kazakh as training data gets close to what we

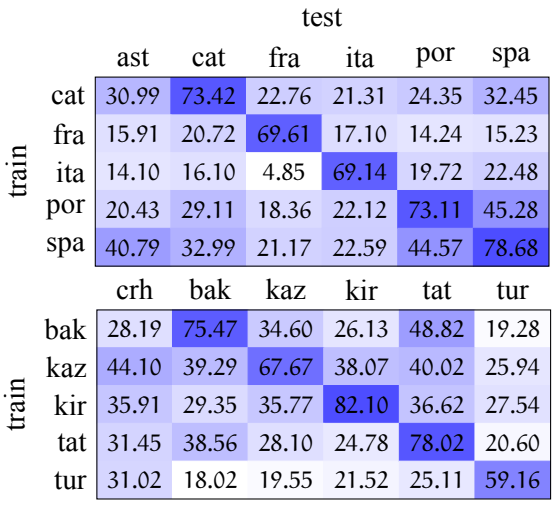

Figure 2: Analysis F1-scores for cross-training languages with another single language in the family: (top) Romance, (bottom) Turkic. All results are obtained using the linear model.

obtained using the complete training data set. It seems the choice of source language(s) is important, and more data, if not appropriate, may even hurt performance depending on the model setup. It is also worth noting that the usefulness of a language as a source language for another exhibits a fair level of asymmetry. Even though the performance matrices presented in Figure 2 (after removing the development set columns) are close to symmetric matrices, there are clear cases of asymmetry as well. For example, using Italian to train a morphological analyzer for French is less useful than using French to train a morphological analyzer for Italian.

Presumably due to distances within the family, French and Italian seem less useful than the other

language in the Romance data set. On Turkic data set, the same seems to be true for Turkish. Excluding Crimean Tatar, Turkish is the least useful language for predicting others. This may also be part of the reason for the difference between the shared task baseline and our systems on the development and test set. Since the baseline system does not transliterate the source languages, it does not benefit from training languages except Turkish. On the other hand, while predicting analysis for the test language Karachay-Balkar, which is written in Cyrillic, the baseline system does not make use of data from Turkish. Not making use of a rather noisy part of the input may in fact be an advantage. Hence, our model outperforms the baseline on the development set by benefiting from all the data. However, for the test language, it gets mislead by a less useful source language that the baseline system simply ignores.

In general, however, the similarity of languages seem to help. The cross-testing results are better for similar languages in Figure 2 in comparison to less-similar ones. In fact, the average performance obtained using language pairs on Romance data correlates highly ($r = 0.83$) with linguistic similarities based on shared cognates (Dellert, 2017), indicating, as expected, usefulness of source languages more similar to the target language.

4.3 Error analysis

In this section, we look at the errors made by the systems on the development set more carefully. As well as reporting the rates of some of the quantifiable aspects of errors, we provide some qualitative analysis of the types of mistakes made by different models.

Most POS tag errors are confusions between POS tags `NOUN` and `VERB`, which may also be largely due to the fact these are also the most frequent POS classes in the data. Otherwise, for both families major confusions are either due to missing some of the ambiguous analyses, or, to a lesser extent, predicting additional (wrong) POS tags. We present confusion tables of POS tags sets of the neural model in Table 8 in Appendix. The tables also show that POS ambiguity is more common in Romance data set.

Given large number of morphological feature–value pairs, a similar analysis is not easy for the morphological features. We count true positive (TP) and false positive (FP) errors, i.e., number of instances of a feature–value pair in gold data missing from the predictions and number of pairs that are predicted but not in the gold data. We present the rates aggregated by each feature label in Table 6 and 7, Turkic and Romance development sets respectively (more detailed versions, reporting error rates for each feature–value pair are presented in Table 10 and 9 in Appendix).

Feature	FP rate	FN rate
`Person[psor]`	0.09	0.07
`Number[psor]`	0.14	0.11
`Case`	0.15	0.14
`Number`	0.28	0.03
`Voice`	0.38	0.20
`Aspect`	0.54	0.10
`Tense`	0.61	0.33
`Valency`	0.62	0.46
`Mood`	0.63	0.50
`VerbForm`	0.71	0.30
`Person`	0.76	0.19
`Deriv`	0.79	0.40
`Missing`	1.00	1.00
`Polarity`	1.00	0.00

Table 6: False positive (FP) and false negative (FN) error rates on feature–value pairs on Turkic development set. The rates are aggregated over the feature label.

Feature	FP rate	FN rate
`Number`	0.03	0.04
`Gender`	0.12	0.13
`Aspect`	0.22	0.14
`VerbForm`	0.24	0.29
`Tense`	0.39	0.31
`Mood`	0.48	0.26
`Person`	0.49	0.24
`Possessive`	1.00	0.00

Table 7: False positive (FP) and false negative (FN) error rates on feature–value pairs on Romance development set. The rates are aggregated over the feature label.

In both families, the nominal features seem to be easier to predict than verbal ones. Besides features that are difficult to interpret, e.g., `Missing` in Turkic data, very high error rates happen with features that are observed only a few times and those with ambiguity. For example, `Possessive` occurs only twice on Asturian data. To exemplify a case with ambiguity of the mapping between the surface strings and the features, we look

at Crimean Tatar suffix *-me*/*-ma*, which is ambiguous between negative and infinitive markers. This ambiguity is the likely cause of complete failure of the model in predicting the `Polarity` features, as well as being responsible for some of the errors for `VerbForm=Vnoun`. We present further (mostly qualitative) error analyses on both development sets below.

ROA Regarding the Asturian development data, both of our models lead to fewer overall predictions than the gold data contains: 1 133 for the linear model and 1 389 for the neural model compared to the 1 461 predictions in the development data, suggesting that our models are conservative when predicting POS tags. This is especially noticeable with the linear model, where 65 % of the POS tag predictions where for `NOUN`. The neural approach gives a similar distribution over POS tags as the gold standard, which suggests that neural models may be better at capturing the ambiguity inherent in morphological prediction.

Both cross-lingual models fail on examples of morphological paradigms that are not found in the training data. An example from Asturian is the formation of the past participles, where the infinitive ending (*-ar*, *-er*, *-ir*) is removed and replaced by the participle ending (*-áu*, *-íu*). Our linear model incorrectly predicts that these are nouns and predicts the same form as the lemma, while the neural model is better able to predict the POS tags, but cannot consistently predict the correct lemma, often choosing a similar lemma from Spanish.

When the POS prediction is correct, the average Levenshtein difference between the predicted and gold lemmas is respectable (0.46 for the linear model, 0.42 for the neural model).

TRK Similar to the ROA development set, both our models make fewer predictions on average than the gold standard predictions provided for Crimean Tatar. As noted in Section 3 the (optimum) linear model makes only a single prediction for each of the 999 word forms. The linear model predicts more with 1 196 analyses in total, close to, but still less than 1 245 gold-standard analyses.

In Turkic development set, systematic errors in lemmatization involve missing multiple lemmas for a form where one of the lemmas is a derived form of another. For example, both models miss the alternative lemma *kiriş* ‘to interfere’ for the word *kirişti* ‘interfered / entered (cooperatively)’, predicting only the simpler form *kir* ‘to enter’. Common prediction errors also include segmenting words at common suffixes. *biznesi* ‘his/her business’ is lemmatized as *bizne*, as *-si* is a common allomorph of the third person singular possessive suffix across Turkic languages, while the loan-word *biznes* is probably an unlikely sequence of letters for a Turkic lemma despite a few occurrences in the training data. Another, possibly fixable, problem for the neural model is due to the letters that do not occur in the training set. For example, the Crimean Tatar data includes the letter *â* which is always predicted as another letter that is most probable in context.

As expected from the overall lemma prediction scores on the Turkic data, when the POS prediction is correct, the average edit difference between the predicted and gold lemmas are lower for the linear model (0.27) than the neural model (0.46).

5 Conclusions

We have presented our submission for the cross-lingual morphological prediction task, which achieved the best tag and analysis scores in the Romance track. We trained both linear and neural morphological analyzers in a pipeline fashion and demonstrated that these models can take advantage of labeled data in source languages to predict the morphological analysis in a similar target language.

While the results presented here are competitive with others obtained in this shared task, the analysis scores are admittedly low. However, there are multiple ways to improve the results as our models do not incorporate much in terms of cross-lingual signal. In the future, it would be worth integrating this cross-lingual signal in the form of pre-trained cross-lingual word embeddings (Artetxe et al., 2016; Lample et al., 2018) or sub-word, e.g., character, embeddings (Chaudhary et al., 2018; Sofroniev and Çöltekin, 2018), as this could lead to better generalization to new languages. Similarly, typological distance between source and target language often correlates with performance (Cotterell and Heigold, 2017), which could be exploited for weighting the contribution of source-language examples when learning a multilingual model.

Acknowledgments

Some of the experiments reported here were run on a Titan Xp donated by the NVIDIA Corporation.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Kemal Altintas and Ilyas Cicekli. 2001. A morphological analyser for Crimean Tatar. In *Proceedings of the 10th Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN'2001)*, pages 180–189.

Carme Armentano-Oller, Rafael C Carrasco, Antonio M Corbí-Bellot, Mikel L Forcada, Mireia Ginestí-Rosell, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, and Miriam A Scalco. 2006. Open-source Portuguese–Spanish machine translation. In *International Workshop on Computational Processing of the Portuguese Language*, pages 50–59. Springer.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2289–2294.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2018. Bilingual sentiment embeddings: Joint projection of sentiment across languages. In *Proceedings of the 56th Annual Meeting of the Association for Comutational Linguistics (Volume 1: Long Papers)*, pages 2483–2493.

Çağrı Çöltekin. 2010. A freely available morphological analyzer for Turkish. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 820–827.

Çağrı Çöltekin and Taraka Rama. 2016. Discriminating similar languages with linear SVMs and neural networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 15–24, Osaka, Japan.

Çağrı Çöltekin and Taraka Rama. 2018. Tübingen-Oslo at SemEval-2018 task 2: SVMs perform better than RNNs at emoji prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 34—38, New Orleans, LA, United States. Association for Computational Linguistics.

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R. Mortensen, and Jaime Carbonell. 2018. Adapting word embeddings to new languages with morphological and phonological subword representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3295. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30. Association for Computational Linguistics.

Johannes Dellert. 2017. *Information-Theoretic Causal Inference of Lexical Flow*. Ph.D. thesis, University of Tübingen.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Manaal Faruqui and Shankar. Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Proceedings of the 2018 Conference

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 1351–1356.

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.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Gulshat Kessikbayeva and Ilyas Cicekli. 2014. Rule based morphological analyzer of Kazakh language. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 46–54.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho Challenge competition 2005–2010: evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Miikka Silfverberg and Francis Tyers. 2019. Data-driven morphological analysis for Uralic languages. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 1–14. Association for Computational Linguistics.

Pavel Sofroniev and Çağrı Çöltekin. 2018. Phonetic vector representations for sound sequence alignment. In *Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, page (to appear).

Francis Tyers, Felipe Sánchez-Martínez, Sergio Ortiz-Rojas, and Mikel Forcada. 2010. Free/open-source resources in the Apertium platform for machine translation research and development. *The Prague Bulletin of Mathematical Linguistics*, 93:67–76.

Evelyne Tzoukermann and Mark Y Liberman. 1990. A finite-state morphological processor for Spanish. In *Proceedings of the 13th conference on Computational linguistics-Volume 3*, pages 277–282. Association for Computational Linguistics.

Jonathan Washington, Ilnar Salimzyanov, and Francis M Tyers. 2014. Finite-state morphological transducers for three Kypchak languages. In *LREC*, pages 3378–3385.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Nat alia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei Butnaru, and Tommi Jauhiainen. 2019. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Association for Computational Linguistics.

A Appendix

	A	J	N	V	AN	JN	JV	NV	ANV	JAN	JNV
A	9	0	0	1	0	1	1	0	1	1	0
J	0	85	1	2	0	37	2	2	0	0	2
N	0	1	237	24	1	42	4	43	1	0	8
V	0	4	12	197	0	4	12	18	0	0	4
AN	0	0	0	0	0	0	0	0	0	1	0
JA	0	1	1	0	0	1	0	0	0	0	0
JN	0	14	14	2	0	102	3	3	0	0	2
JV	0	1	1	7	0	4	12	4	0	0	1
NV	0	1	13	3	0	0	0	31	0	0	0
JNV	0	0	2	3	0	4	1	7	0	0	4

	A	J	N	V	AN	JA	JN	JV	NV
A	0	0	2	1	1	0	0	0	0
J	0	8	18	2	0	0	13	0	5
N	0	2	662	15	1	1	15	0	24
V	0	3	51	116	1	0	4	3	25
AN	0	0	3	0	1	0	0	0	0
JA	0	0	1	0	0	0	0	0	0
JN	0	0	2	0	1	0	0	0	0
JV	0	1	0	2	0	0	0	0	0
NV	0	0	5	2	0	0	2	0	6

Table 8: Confusion matrix for Asturian (left) and Crimean Tatar (right) data sets for all POS combinations. The letters in the column and row labels are adverb (A) adjective (J), noun (N) and verb (V), where combination of letters indicate words that are assigned all indicated POS tags in the gold standard (rows) or predictions (columns). Columns and rows with all zeros were removed.

Feature=Value	FP	NP	FP rate	FN	NN	FN rate
`VerbForm=Ger`	0	22	0.00	0	22	0.00
`VerbForm=Inf`	0	48	0.00	4	52	0.08
`Number=Plur`	6	324	0.02	4	322	0.01
`Number=Sing`	22	619	0.04	27	624	0.04
`Gender=Fem`	28	326	0.09	28	326	0.09
`Gender=Masc`	44	363	0.12	53	372	0.14
`Aspect=Perf`	7	56	0.13	11	60	0.18
`VerbForm=Fin`	35	198	0.18	60	223	0.27
`Gender=Masc,Fem`	20	109	0.18	21	110	0.19
`Person=3`	41	161	0.25	22	142	0.15
`Tense=Past`	57	167	0.34	48	158	0.30
`Mood=Ind`	66	173	0.38	32	139	0.23
`Aspect=Imp`	11	27	0.41	0	16	0.00
`Tense=Pres`	50	108	0.46	26	84	0.31
`Mood=Sub`	24	50	0.48	15	41	0.37
`Mood=Cnd`	3	6	0.50	0	3	0.00
`Number=Sing,Plur`	2	4	0.50	4	6	0.67
`Mood=Imp`	19	36	0.53	8	25	0.32
`VerbForm=Part`	51	87	0.59	46	82	0.56
`Person=1`	23	39	0.59	2	18	0.11
`Person=2`	48	67	0.72	24	43	0.56
`Possessive=Yes`	2	2	1.00	0	0	0.00

Table 9: False positive (FP) and false negative (FN) error rates of the neural model on the Romance development set (Asturian). NP indicate number of instance of the feature-value pair in the gold data, NN indicate the total number of instances in the predictions.

Feature=Value	FP	NP	FP rate	FN	NN	FN rate
Case=Abl	0	50	0.00	31	81	0.38
Case=Gen	0	79	0.00	0	79	0.00
Case=Loc	0	82	0.00	3	85	0.04
Case=Acc	2	79	0.03	6	83	0.07
Person[psor]=3	17	356	0.05	25	364	0.07
Number=Plur	12	225	0.05	1	214	0.00
Number[psor]=Sing,Plur	32	356	0.09	23	347	0.07
Case=Nom	76	441	0.17	76	441	0.17
Case=Dat	20	106	0.19	5	91	0.05
Case=Sim	2	6	0.33	0	4	0.00
Voice=Pass	21	56	0.38	9	44	0.20
Valency=2	54	140	0.39	74	160	0.46
Tense=Past	38	79	0.48	3	44	0.07
Aspect=Imp	14	28	0.50	1	15	0.07
Aspect=Perf	37	68	0.54	4	35	0.11
VerbForm=Fin	67	119	0.56	20	72	0.28
VerbForm=Conv	49	80	0.61	3	34	0.09
Mood=Imp	10	14	0.71	4	8	0.50
Person=3	75	102	0.74	5	32	0.16
Number=Sing	81	107	0.76	7	33	0.21
Deriv=Coop	11	14	0.79	2	5	0.40
VerbForm=Vnoun	88	110	0.80	7	29	0.24
Tense=Aor	34	42	0.81	12	20	0.60
Person[psor]=1	15	18	0.83	0	3	0.00
VerbForm=Part	37	44	0.84	2	9	0.22
Person=2	14	16	0.88	2	4	0.50
Valency=1	87	95	0.92	5	13	0.38
Aspect=Prog	2	2	1.00	0	0	0.00
Case=Ins	29	29	1.00	1	1	1.00
Missing=ger_abst	21	21	1.00	1	1	1.00
Missing=ger_fut	3	3	1.00	0	0	0.00
Number[psor]=Plur	7	7	1.00	7	7	1.00
Number[psor]=Sing	13	13	1.00	12	12	1.00
Person=1	2	2	1.00	0	0	0.00
Person[psor]=2	2	2	1.00	2	2	1.00
Polarity=Neg	22	22	1.00	0	0	0.00
Tense=Fut	5	5	1.00	0	0	0.00

Table 10: False positive (FP) and false negative (FN) error rates of the neural model on the Turkic development set (Crimean Tatar). NP indicate number of instance of the feature-value pair in the gold data, NN indicate the total number of instances in the predictions.