Neural Factor Graph Models for Cross-lingual Morphological Tagging

Chaitanya Malaviya and **Matthew R. Gormley** and **Graham Neubig**
Language Technologies Institute, Machine Learning Department
Carnegie Mellon University
{cmalaviy,mgormley,gneubig}@cs.cmu.edu

Abstract

Morphological analysis involves predicting the syntactic traits of a word (e.g. {*POS: Noun, Case: Acc, Gender: Fem*}). Previous work in morphological tagging improves performance for low-resource languages (LRLs) through cross-lingual training with a high-resource language (HRL) from the same family, but is limited by the strict—often false—assumption that tag sets exactly overlap between the HRL and LRL. In this paper we propose a method for cross-lingual morphological tagging that aims to improve information sharing between languages by relaxing this assumption. The proposed model uses factorial conditional random fields with neural network potentials, making it possible to (1) utilize the expressive power of neural network representations to smooth over superficial differences in the surface forms, (2) model pairwise and transitive relationships between tags, and (3) accurately generate tag sets that are unseen or rare in the training data. Experiments on four languages from the Universal Dependencies Treebank (Nivre et al., 2017) demonstrate superior tagging accuracies over existing cross-lingual approaches.[1]

1 Introduction

Morphological analysis (Hajič and Hladká (1998), Oflazer and Kuruöz (1994), *inter alia*) is the task of predicting fine-grained annotations about the syntactic properties of tokens in a language such
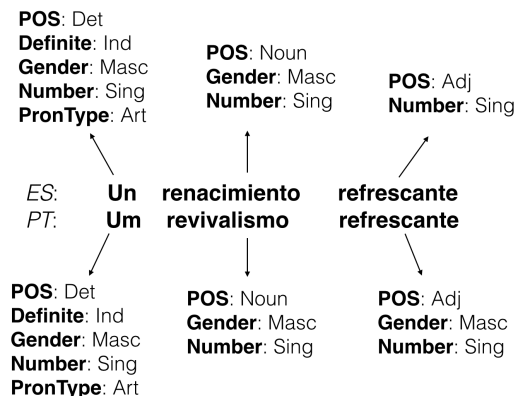

Figure 1: Morphological tags for a UD sentence in Portuguese and a translation in Spanish

as part-of-speech, case, or tense. For instance, in Figure 1, the given Portuguese sentence is labeled with the respective morphological tags such as Gender and its label value Masculine.

The accuracy of morphological analyzers is paramount, because their results are often a first step in the NLP pipeline for tasks such as translation (Vylomova et al., 2017; Tsarfaty et al., 2010) and parsing (Tsarfaty et al., 2013), and errors in the upstream analysis may cascade to the downstream tasks. One difficulty, however, in creating these taggers is that only a limited amount of annotated data is available for a majority of the world's languages to learn these morphological taggers. Fortunately, recent efforts in morphological annotation follow a standard annotation schema for these morphological tags across languages, and now the Universal Dependencies Treebank (Nivre et al., 2017) has tags according to this schema in 60 languages.

Cotterell and Heigold (2017) have recently shown that combining this shared schema with cross-lingual training on a related high-resource language (HRL) gives improved performance

---

[1] Our code and data is publicly available at www.github.com/chaitanyamalaviya/NeuralFactorGraph.
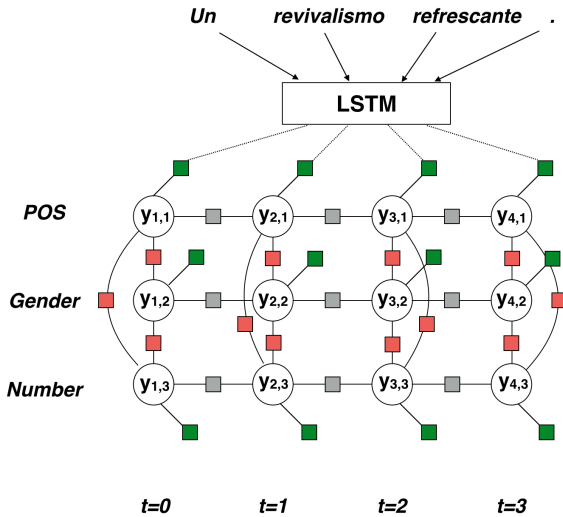
Figure 2: FCRF-LSTM Model for morphological tagging

on tagging accuracy for low-resource languages (LRLs). The output space of this model consists of tag sets such as {*POS: Adj, Gender: Masc, Number: Sing*}, which are predicted for a token at each time step. However, this model relies heavily on the fact that *the entire space of tag sets* for the LRL must match those of the HRL, which is often not the case, either due to linguistic divergence or small differences in the annotation schemes between the two languages.[2] For instance, in Figure 1 "refrescante" is assigned a gender in the Portuguese UD treebank, but not in the Spanish UD treebank.

In this paper, we propose a method that instead of predicting full tag sets, makes predictions over single tags separately but ties together each decision by modeling variable dependencies between tags over time steps (e.g. capturing the fact that nouns frequently occur after determiners) and pairwise dependencies between all tags at a single time step (e.g. capturing the fact that infinitive verb forms don't have tense). The specific model is shown in Figure 2, consisting of a factorial conditional random field (FCRF; Sutton et al. (2007)) with neural network potentials calculated by long short-term memory (LSTM; (Hochreiter and Schmidhuber, 1997)) at every variable node (§3). Learning and inference in the model is made

---

tractable through belief propagation over the possible tag combinations, allowing the model to consider an exponential label space in polynomial time (§3.5).

This model has several advantages:

- The model is able to generate tag sets unseen in training data, and share information between similar tag sets, alleviating the main disadvantage of previous work cited above.

- Our model is empirically strong, as validated in our main experimental results: it consistently outperforms previous work in cross-lingual low-resource scenarios in experiments.

- Our model is more interpretable, as we can probe the model parameters to understand which variable dependencies are more likely to occur in a language, as we demonstrate in our analysis.

In the following sections, we describe the model and these results in more detail.

2   Problem Formulation and Baselines

2.1   Problem Formulation

Formally, we define the problem of morphological analysis as the task of mapping a length-T string of tokens $\mathbf{x} = x_1, \ldots, x_T$ into the target morphological tag sets for each token $\mathbf{y} = \mathbf{y}_1, \ldots, \mathbf{y}_T$. For the tth token, the target label $\mathbf{y}_t = y_{t,1}, \ldots, y_{t,m}$ defines a set of tags (e.g. {*Gender: Masc, Number: Sing, POS: Verb*}). An annotation schema defines a set $\mathcal{S}$ of M possible tag types and with the mth type (e.g. *Gender*) defining its set of possible labels $\mathcal{Y}_m$ (e.g. {*Masc, Fem, Neu*}) such that $y_{t,m} \in \mathcal{Y}_m$. We must note that not all tags or attributes need to be specified for a token; usually, a subset of $\mathcal{S}$ is specified for a token and the remaining tags can be treated as mapping to a NULL $\in \mathcal{Y}_m$ value. Let $\mathcal{Y} = \{(y_1, \ldots, y_M) : y_1 \in \mathcal{Y}_1, \ldots, y_M \in \mathcal{Y}_M\}$ denote the set of all possible tag sets.

2.2   Baseline: Tag Set Prediction

Data-driven models for morphological analysis are constructed using training data $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$ consisting of N training examples. The baseline model (Cotterell and Heigold, 2017) we compare with regards the output space of the model as a subset $\tilde{\mathcal{Y}} \subset \mathcal{Y}$ where $\tilde{\mathcal{Y}}$ is the

set of all tag sets seen in this training data. Specifically, they solve the task as a multi-class classification problem where the classes are individual tag sets. In low-resource scenarios, this indicates that $|\tilde{\mathcal{Y}}| << |\mathcal{Y}|$ and even for those tag sets existing in $\tilde{\mathcal{Y}}$ we may have seen very few training examples. The conditional probability of a sequence of tag sets given the sentence is formulated as a 0th order CRF.

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}) \quad (1)$$

Instead, we would like to be able to generate any combination of tags from the set $\mathcal{Y}$, and share statistical strength among similar tag sets.

2.3 A Relaxation: Tag-wise Prediction

As an alternative, we could consider a model that performs prediction for each tag's label $y_{t,m}$ independently.

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} \prod_{m=1}^{M} p(y_{t,m}|\mathbf{x}) \quad (2)$$

This formulation has an advantage: the tag-predictions within a single time step are now independent, it is now easy to generate any combination of tags from $\mathcal{Y}$. On the other hand, now it is difficult to model the interdependencies between tags in the same tag set $\mathbf{y}_i$, a major disadvantage over the previous model. In the next section, we describe our proposed neural factor graph model, which can model not only dependencies within tags for a single token, but also dependencies across time steps while still maintaining the flexibility to generate any combination of tags from $\mathcal{Y}$.

3 Neural Factor Graph Model

Due to the correlations between the syntactic properties that are represented by morphological tags, we can imagine that capturing the relationships between these tags through pairwise dependencies can inform the predictions of our model. These dependencies exist both among tags for the same token (intra-token pairwise dependencies), and across tokens in the sentence (inter-token transition dependencies). For instance, knowing that a token's POS tag is a Noun, would strongly suggest that this token would have a NULL label for the tag Tense, with very few exceptions (Nordlinger and

Sadler, 2004). In a language where nouns follow adjectives, a tag set prediction {*POS: Adj, Gender: Fem*} might inform the model that the next token is likely to be a noun and have the same gender. The baseline model can not explicitly model such interactions given their factorization in equation 1.

To incorporate the dependencies discussed above, we define a factorial CRF (Sutton et al., 2007), with pairwise links between cotemporal variables and transition links between the same types of tags. This model defines a distribution over the tag-set sequence $\mathbf{y}$ given the input sentence $\mathbf{x}$ as,

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \prod_{\alpha \in \mathcal{C}} \psi_\alpha(\mathbf{y}_\alpha, \mathbf{x}, t) \quad (3)$$

where $\mathcal{C}$ is the set of factors in the factor graph (as shown in Figure 2), α is one such factor, and $\mathbf{y}_\alpha$ is the assignment to the subset of variables neighboring factor α. We define three types of potential functions: neural ψ_{NN}, pairwise ψ_P, and transition ψ_T, described in detail below.
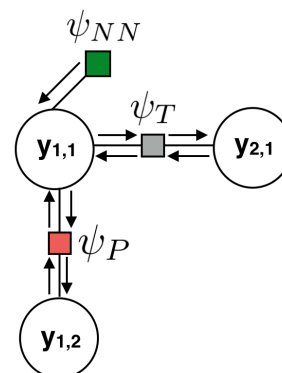

Figure 3: Factors in the Neural Factor Graph model (red: Pairwise, grey: Transition, green: Neural Network)

3.1 Neural Factors

The flexibility of our formulation allows us to include any form of custom-designed potentials in our model. Those for the neural factors have a fairly standard log-linear form,

$$\psi_{NN,i}(y_{t,m}) = \exp\left\{\sum_k \lambda_{\text{nn},k} f_{\text{nn},k}(\mathbf{x}, t)\right\} \quad (4)$$

except that the features $f_{\text{nn},k}$ are themselves given by a neural network. There is one such factor per

variable. We obtain our neural factors using a biLSTM over the input sequence $\mathbf{x}$, where the input word embedding for each token is obtained from a character-level biLSTM embedder. This component of our model is similar to the model proposed by Cotterell and Heigold (2017). Given an input token $\mathbf{x}_t = c_1...c_n$, we compute an input embedding v_t as,

$$v_t = [\text{cLSTM}(c_1...c_n); \text{cLSTM}(c_n...c_1)] \quad (5)$$

Here, cLSTM is a character-level LSTM function that returns the last hidden state. This input embedding v_t is then used in the biLSTM tagger to compute an output representation e_t. Finally, the scores $f_{\text{nn}}(\mathbf{x}, t)$ are obtained as,

$$f_{\text{nn}}(\mathbf{x}, t) = W_l e_t + b_l \quad (6)$$

We use a language-specific linear layer with weights W_l and bias b_l.

3.2 Pairwise Factors

As discussed previously, the pairwise factors are crucial for modeling correlations between tags. The pairwise factor potential for a tag i and tag j at timestep t is given in equation 7. Here, the dimension of f_p is $(|\mathcal{Y}_i|, |\mathcal{Y}_j|)$. These scores are used to define the neural factors as,

$$\psi_{P_{i,j}}(y_{t,i}, y_{t,j}) = \exp\left\{ \sum_k \lambda_{p,k} f_{p,k}(y_{t,i}, y_{t,j}) \right\} \quad (7)$$

3.3 Transition Factors

Previous work has experimented with the use of a linear chain CRF with factors from a neural network (Huang et al., 2015) for sequence tagging tasks. We hypothesize that modeling transition factors in a similar manner can allow the model to utilize information about neighboring tags and capture word order features of the language. The transition factor for tag i and timestep t is given below for variables $y_{t,i}$ and $y_{t+1,i}$. The dimension of f_T is $(|\mathcal{Y}_i|, |\mathcal{Y}_i|)$.

$$\psi_{T_{i,t}}(y_{t,i}, y_{t+1,i}) = \exp\left\{ \sum_k \lambda_{T,k} f_{T,k}(y_{t,i}, y_{t+1,i}) \right\} \quad (8)$$

In our experiments, $f_{p,k}$ and $f_{T,k}$ are simple indicator features for the values of tag variables with no dependence on $\mathbf{x}$.

3.4 Language-Specific Weights

As an enhancement to the information encoded in the transition and pairwise factors, we experiment with training general and language-specific parameters for the transition and the pairwise weights. We define the weight matrix λ_{gen} to learn the general trends that hold across both languages, and the weights λ_{lang} to learn the exceptions to these trends. In our model, we sum both these parameter matrices before calculating the transition and pairwise factors. For instance, the transition weights λ_T are calculated as $\lambda_T = \lambda_{\text{T, gen}} + \lambda_{\text{T, lang}}$.

3.5 Loopy Belief Propagation

Since the graph from Figure 2 is a loopy graph, performing exact inference can be expensive. Hence, we use loopy belief propagation (Murphy et al., 1999; Ihler et al., 2005) for computation of approximate variable and factor marginals. Loopy BP is an iterative message passing algorithm that sends messages between variables and factors in a factor graph. The message updates from variable v_i, with neighboring factors $N(i)$, to factor α is

$$\mu_{i \rightarrow \alpha}(v_i) = \prod_{\alpha \in N(i) \backslash \alpha} \mu_{\alpha \rightarrow i}(v_i) \quad (9)$$

The message from factor α to variable v_i is

$$\mu_{\alpha \rightarrow i}(v_i) = \sum_{\mathbf{v}_\alpha : \mathbf{v}_\alpha[i] = v_i} \psi_\alpha(\mathbf{v}_\alpha) \prod_{j \in N(\alpha) \backslash i} \mu_{j \rightarrow \alpha}(\mathbf{v}_\alpha[i]) \quad (10)$$

where $\mathbf{v}_\alpha$ denote an assignment to the subset of variables adjacent to factor α, and $\mathbf{v}_\alpha[i]$ is the assignment for variable v_i. Message updates are performed asynchronously in our model. Our message passing schedule was similar to that of foward-backward: the forward pass sends all messages from the first time step in the direction of the last. Messages to/from pairwise factors are included in this forward pass. The backward pass sends messages in the direction from the last time step back to the first. This process is repeated until convergence. We say that BP has converged when the maximum residual error (Sutton and McCallum, 2007) over all messages is below some threshold. Upon convergence, we obtain the belief values of variables and factors as,

$$b_i(v_i) = \frac{1}{\kappa_i} \prod_{\alpha \in N(i)} \mu_{\alpha \rightarrow i}(v_i) \quad (11)$$

$$b_\alpha(v_\alpha) = \frac{1}{\kappa_\alpha} \psi_\alpha(v_\alpha) \prod_{i \in N(\alpha)} \mu_{i \rightarrow \alpha}(v_\alpha[i]) \quad (12)$$

where κ_i and κ_α are normalization constants ensuring that the beliefs for a variable i and factor α sum-to-one. In this way, we can use the beliefs as approximate marginal probabilities.

3.6 Learning and Decoding

We perform end-to-end training of the neural factor graph by following the (approximate) gradient of the log-likelihood $\sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$. The true gradient requires access to the marginal probabilities for each factor, e.g. $p(\mathbf{y}_\alpha|\mathbf{x})$ where $\mathbf{y}_\alpha$ denotes the subset of variables in factor α. For example, if α is a transition factor for tag m at timestep t, then $\mathbf{y}_\alpha$ would be $y_{t,m}$ and $y_{t+1,m}$. Following (Sutton et al., 2007), we replace these marginals with the beliefs $b_\alpha(\mathbf{y}_\alpha)$ from loopy belief propagation.[3] Consider the log-likelihood of a single example $\ell^{(i)} = \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$. The partial derivative with respect to parameter $\lambda_{g,k}$ for each type of factor $g \in \{NN, T, P\}$ is the difference of the observed features with the expected features under the model's (approximate) distribution as represented by the beliefs:

$$\frac{\partial \ell^{(i)}}{\partial \lambda_{g,k}} = \sum_{\alpha \in \mathcal{C}_g} \left(f_{g,k}(\mathbf{y}_\alpha^{(i)}) - \sum_{\mathbf{y}_\alpha} b_\alpha(\mathbf{y}_\alpha) f_{g,k}(\mathbf{y}_\alpha) \right)$$

where $\mathcal{C}_g$ denotes all the factors of type g, and we have omitted any dependence on $\mathbf{x}^{(i)}$ and t for brevity—t is accessible through the factor index α. For the neural network factors, the features are given by a biLSTM. We backpropagate through to the biLSTM parameters using the partial derivative below,

$$\frac{\partial \ell^{(i)}}{\partial f_{NN,k}(y_{t,m}^{(i)}, t)} = \lambda_{NN,k} - \sum_{y_{t,m}} b_{t,m}(y_{t,m})\lambda_{NN,k}$$

where $b_{t,m}(\cdot)$ is the variable belief corresponding to variable $y_{t,m}$.

To predict a sequence of tag sets $\hat{\mathbf{y}}$ at test time, we use minimum Bayes risk (MBR) decoding (Bickel and Doksum, 1977; Goodman, 1996) for Hamming loss over tags. For a variable $y_{t,m}$ representing tag m at timestep t, we take

$$\hat{y}_{t,m} = \arg\max_{l \in \mathcal{Y}_m} b_{t,m}(l). \tag{13}$$

where l ranges over the possible labels for tag m.

Language Pair	HRL Train	Dev	Test
DA/SV	4,383	504	1219
RU/BG	3,850	1115	1116
FI/HU	12,217	441	449
ES/PT	14,187	560	477

Table 1: Dataset sizes. $tgt\_size = 100$ or $1,000$ LRL sentences are added to HRL Train

Language Pair	Unique Tags	Tag Sets
DA/SV	23	224
RU/BG	19	798
FI/HU	27	2195
ES/PT	19	451

Table 2: Tag Set Sizes with tgt_size=100

4 Experimental Setup

4.1 Dataset

We used the Universal Dependencies Treebank UD v2.1 (Nivre et al., 2017) for our experiments. We picked four low-resource/high-resource language pairs, each from a different family: Danish/Swedish (DA/SV), Russian/Bulgarian (RU/BG), Finnish/Hungarian (FI/HU), Spanish/Portuguese (ES/PT). Picking languages from different families would ensure that we obtain results that are on average consistent across languages.

The sizes of the training and evaluation sets are specified in Table 1. In order to simulate low-resource settings, we follow the experimental procedure from Cotterell and Heigold (2017). We restrict the number of sentences of the target language (tgt_size) in the training set to 100 or 1000 sentences. We also augment the tag sets in our training data by adding a NULL label for all tags that are not seen for a token. It is expected that our model will learn which tags are unlikely to occur given the variable dependencies in the factor graph. The dev set and test set are only in the target language. From Table 2, we can see there is also considerable variance in the number of unique tags and tag sets found in each of these language pairs.

---

[3]Using this approximate gradient is akin to the *surrogate likelihood* training of (Wainwright, 2006).

Language	Model	$tgt\_size = 100$			$tgt\_size=1000$		
		Accuracy	F1-Micro	F1-Macro	Accuracy	F1-Macro	F1-Micro
SV	Baseline	15.11	8.36	10.37	68.64	76.36	76.50
	Ours	**29.47**	**54.09**	**54.36**	**71.32**	**84.42**	**84.46**
BG	Baseline	**29.05**	14.32	29.62	**59.20**	**67.22**	**67.12**
	Ours	27.81	**40.97**	**42.43**	39.25	60.23	60.84
HU	Baseline	21.97	13.30	16.67	**50.75**	58.68	62.79
	Ours	**33.32**	**54.88**	**54.69**	45.90	**74.05**	**73.38**
PT	Baseline	18.91	7.10	10.33	74.22	81.62	81.87
	Ours	**58.82**	**73.67**	**74.07**	**76.26**	**87.13**	**87.22**

Table 3: Token-wise accuracy and F1 scores on mono-lingual experiments

4.2 Baseline Tagger

As the baseline tagger model, we re-implement the SPECIFIC model from Cotterell and Heigold (2017) that uses a language-specific softmax layer. Their model architecture uses a character biLSTM embedder to obtain a vector representation for each token, which is used as input in a word-level biLSTM. The output space of their model is all the tag sets seen in the training data. This work achieves strong performance on several languages from UD on the task of morphological tagging and is a strong baseline.

4.3 Training Regimen

We followed the parameter settings from Cotterell and Heigold (2017) for the baseline tagger and the neural component of the FCRF-LSTM model. For both models, we set the input embedding and linear layer dimension to 128. We used 2 hidden layers for the LSTM where the hidden layer dimension was set to 256 and a dropout (Srivastava et al., 2014) of 0.2 was enforced during training. All our models were implemented in the PyTorch toolkit (Paszke et al., 2017). The parameters of the character biLSTM and the word biLSTM were initialized randomly. We trained the baseline models and the neural factor graph model with SGD and Adam respectively for 10 epochs each, in batches of 64 sentences. These optimizers gave the best performances for the respective models.

For the FCRF, we initialized transition and pairwise parameters with zero weights, which was important to ensure stable training. We considered BP to have reached convergence when the maximum residual error was below 0.05 or if the maximum number of iterations was reached (set to 40 in our experiments). We found that in cross-lingual experiments, when $tgt\_size = 100$, the relatively large amount of data in the HRL was causing our model to overfit on the HRL and not generalize well to the LRL. As a solution to this, we upsampled the LRL data by a factor of 10 when $tgt\_size = 100$ for both the baseline and the proposed model.

Evaluation: Previous work on morphological analysis (Cotterell and Heigold, 2017; Buys and Botha, 2016) has reported scores on average token-level accuracy and F1 measure. The average token level accuracy counts a tag set prediction as correct only it is an exact match with the gold tag set. On the other hand, F1 measure is measured on a tag-by-tag basis, which allows it to give partial credit to partially correct tag sets. Based on the characteristics of each evaluation measure, Accuracy will favor tag-set prediction models (like the baseline), and F1 measure will favor tag-wise prediction models (like our proposed method). Given the nature of the task, it seems reasonable to prefer getting some of the tags correct (e.g. Noun+Masc+Sing becomes Noun+Fem+Sing), instead of missing all of them (e.g. Noun+Masc+Sing becomes Adj+Fem+Plur). F-score gives partial credit for getting some of the tags correct, while tagset-level accuracy will treat these two mistakes equally. Based on this, we believe that F-score is intuitively a better metric. However, we report both scores for completeness.

5 Results and Analysis

5.1 Main Results

First, we report the results in the case of mono-lingual training in Table 3. The first row for each language pair reports the results for our reimple-

Language	Model	$tgt\_size = 100$			$tgt\_size = 1000$		
		Accuracy	F1-Micro	F1-Macro	Accuracy	F1-Macro	F1-Micro
DA/SV	Baseline	**66.06**	73.95	74.37	**82.26**	**87.88**	**87.91**
	Ours	63.22	**78.75**	**78.72**	77.43	87.56	87.52
RU/BG	Baseline	**52.76**	58.41	58.23	**71.90**	77.89	77.97
	Ours	46.89	**64.46**	**64.75**	67.56	**82.06**	**82.11**
FI/HU	Baseline	**51.74**	68.15	66.82	61.80	75.96	76.16
	Ours	45.41	**68.63**	**68.07**	**63.93**	**85.06**	**84.12**
ES/PT	Baseline	**79.40**	86.03	86.14	**85.85**	91.91	91.93
	Ours	77.75	**88.42**	**88.44**	85.02	**92.35**	**92.37**

Table 4: Token-wise accuracy and F1 scores on cross-lingual experiments

mentation of Cotterell and Heigold (2017), and the second for our full model. From these results, we can see that we obtain improvements on the F-measure over the baseline method in most experimental settings except BG with $tgt\_size = 1000$. In a few more cases, the baseline model sometimes obtains higher accuracy scores for the reason described in 4.3.

In our cross-lingual experiments shown in Table 4, we also note F-measure improvements over the baseline model with the exception of DA/SV when $tgt\_size = 1000$. We observe that the improvements are on average stronger when $tgt\_size = 100$. This suggests that our model performs well with very little data due to its flexibility to generate any tag set, including those not observed in the training data. The strongest improvements are observed for FI/HU. This is likely because the number of unique tags is the highest in this language pair and our method scales well with the number of tags due to its ability to make use of correlations between the tags in different tag sets.

factors completely from the model in Table 5. Ablation experiments for each factor showed decreases in scores relative to the model where both factors are present, but the decrease attributed to the pairwise factors is larger, in both the monolingual and cross-lingual cases. Removing both factors from our proposed model results in a further decrease in the scores. These differences were found to be more significant in the case when $tgt\_size = 100$.

Upon looking at the tag set predictions made by our model, we found instances where our model utilizes variable dependencies to predict correct labels. For instance, for a specific phrase in Portuguese (*um estado*), the baseline model predicted {*POS: Det, Gender: Masc, Number: Sing*}$_t$, {*POS: Noun, Gender: Fem (X), Number: Sing*}$_{t+1}$, whereas our model was able to get the gender correct because of the transition factors in our model.

5.2 What is the Model Learning?

Language	Transition	Pairwise	F1-Macro
HU	×	×	69.87
	✓	×	73.21
	×	✓	73.68
	✓	✓	74.05
FI/HU	×	×	79.57
	✓	×	84.41
	×	✓	84.73
	✓	✓	85.06

Table 5: Ablation Experiments ($tgt\_size = 1000$)

To examine the utility of our transition and pairwise factors, we also report results on ablation experiments by removing transition and pairwise
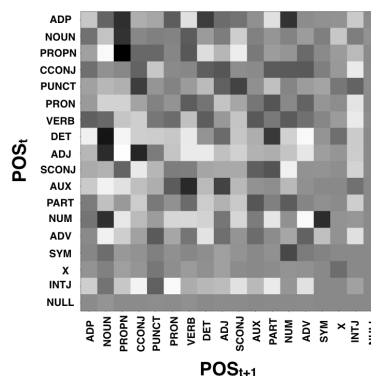

Figure 4: Generic transition weights for POS from the RU/BG model

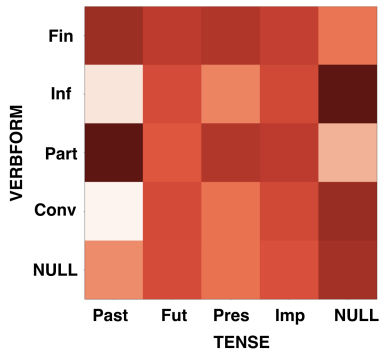One of the major advantages of our model is

Figure 5: Generic pairwise weights between `Verbform` and `Tense` from the RU/BG model

the ability to interpret what the model has learned by looking at the trained parameter weights. We investigated both language-generic and language-specific patterns learned by our parameters:

- **Language-Generic**: We found evidence for several syntactic properties learned by the model parameters. For instance, in Figure 4, we visualize the generic ($\lambda_{T, gen}$) transition weights of the POS tags in RU/BG. Several universal trends such as determiners and adjectives followed by nouns can be seen. In Figure 5, we also observed that infinitive has a strong correlation for `NULL` tense, which follows the universal phenomena that infinitives don't have tense.
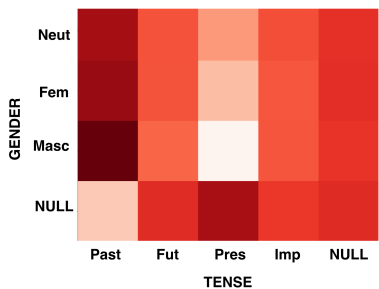

Figure 6: Language-specific pairwise weights for RU between `Gender` and `Tense` from the RU/BG model

- **Language Specific Trends**: We visualized the learnt language-specific weights and looked for evidence of patterns corresponding to linguistic phenomenas observed in a language of interest. For instance, in Russian, verbs are gender-specific in past tense but not in other tenses. To analyze this, we plotted pairwise weights for Gender/Tense in

Figure 6 and verified strong correlations between the past tense and all gender labels.

6 Related Work

There exist several variations of the task of prediction of morphological information from annotated data: paradigm completion (Durrett and DeNero, 2013; Cotterell et al., 2017b), morphological reinflection (Cotterell et al., 2017a), segmentation (Creutz et al., 2005; Cotterell et al., 2016) and tagging. Work on morphological tagging has broadly focused on structured prediction models such as CRFs, and neural network models. Amongst structured prediction approaches, Müller et al. (2013); Müller and Schütze (2015) proposed the use of a higher-order CRF that is approximated using coarse-to-fine decoding. (Müller et al., 2015) proposed joint lemmatization and tagging using this framework. (Hajič, 2000) was the first work that performed experiments on multilingual morphological tagging. They proposed an exponential model and the use of a morphological dictionary. Buys and Botha (2016); Kirov et al. (2017) proposed a model that used tag projection of type and token constraints from a resource-rich language to a low-resource language for tagging.

Most recent work has focused on character-based neural models (Heigold et al., 2017), that can handle rare words and are hence more useful to model morphology than word-based models. These models first obtain a character-level representation of a token from a biLSTM or CNN, which is provided to a word-level biLSTM tagger. Heigold et al. (2017, 2016) compared several neural architectures to obtain these character-based representations and found the effect of the neural network architecture to be minimal given the networks are carefully tuned. Cross-lingual transfer learning has previously boosted performance on tasks such as translation (Johnson et al., 2016) and POS tagging (Snyder et al., 2008; Plank et al., 2016). Cotterell and Heigold (2017) proposed a cross-lingual character-level neural morphological tagger. They experimented with different strategies to facilitate cross-lingual training: a language ID for each token, a language-specific softmax and a joint language identification and tagging model. We have used this work as a baseline model for comparing with our proposed method.

In contrast to earlier work on morphological tagging, we use a hybrid of neural and graphical

model approaches. This combination has several advantages: we can make use of expressive feature representations from neural models while ensuring that our model is interpretable. Our work is similar in spirit to Huang et al. (2015) and Ma and Hovy (2016), who proposed models that use a CRF with features from neural models. For our graphical model component, we used a factorial CRF (Sutton et al., 2007), which is a generalization of a linear chain CRF with additional pairwise factors between cotemporal variables.

7 Conclusion and Future Work

In this work, we proposed a novel framework for sequence tagging that combines neural networks and graphical models, and showed its effectiveness on the task of morphological tagging. We believe this framework can be extended to other sequence labeling tasks in NLP such as semantic role labeling. Due to the robustness of the model across languages, we believe it can also be scaled to perform morphological tagging for multiple languages together.

Acknowledgments

References

Peter J. Bickel and Kjell A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day Inc., Oakland, CA, USA.

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1954–1964.

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*. Association for Computational Linguistics, Copenhagen, Denmark, pages 748–759.

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

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2016. Morphological segmentation inside-out. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2325–2330.

Ryan Cotterell, Ekaterina Vylomova, Huda Khayrallah, Christo Kirov, and David Yarowsky. 2017b. Paradigm completion for derivational morphology. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 714–720.

Mathias Creutz, Krista Lagus, Krister Lindén, and Sami Virpioja. 2005. Morfessor and hutmegs: Unsupervised morpheme segmentation for highly-inflecting and compounding languages .

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1185–1195.

Joshua Goodman. 1996. Efficient algorithms for parsing the DOP model. In *Proceedings of EMNLP*.

Jan Hajič. 2000. Morphological tagging: Data vs. dictionaries. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, pages 94–101.

Jan Hajič and Barbora Hladká. 1998. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 483–490.

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2016. Neural morphological tagging from characters for morphologically rich languages. *arXiv preprint arXiv:1606.06640* .

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 505–513.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

Alexander T Ihler, W Fisher John III, and Alan S Willsky. 2005. Loopy belief propagation: Convergence and effects of message errors. *Journal of Machine Learning Research* 6(May):905–936.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558* .

Christo Kirov, John Sylak-Glassman, Rebecca Knowles, Ryan Cotterell, and Matt Post. 2017. A rich morphological tagger for english: Exploring the cross-linguistic tradeoff between morphology and syntax. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. volume 2, pages 112–117.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1064–1074.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 2268–2274.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 322–332.

Thomas Müller and Hinrich Schütze. 2015. Robust morphological tagging with word representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 526–536.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pages 467–475.

Joakim Nivre et al. 2017. Universal dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rachel Nordlinger and Louisa Sadler. 2004. Nominal tense in crosslinguistic perspective. *Language* 80(4):776–806.

Kemal Oflazer and Ilker Kuruöz. 1994. Tagging and morphological disambiguation of turkish text. In *Proceedings of the fourth conference on Applied natural language processing*. Association for Computational Linguistics, pages 144–149.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch .

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 412–418.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2008. Unsupervised multilingual learning for pos tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1041–1050.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Charles Sutton and Andrew McCallum. 2007. Improved dynamic schedules for belief propagation. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research* 8(Mar):693–723.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl): what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*. Association for Computational Linguistics, pages 1–12.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologically rich languages: Introduction to the special issue. *Computational linguistics* 39(1):15–22.

Ekaterina Vylomova, Trevor Cohn, Xuanli He, and Gholamreza Haffari. 2017. Word representation models for morphologically rich languages in neural machine translation. In *Proceedings of the First*

Workshop on Subword and Character Level Models in NLP. Association for Computational Linguistics, Copenhagen, Denmark, pages 103–108.

Martin J Wainwright. 2006. Estimating the"wrong"graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research* 7(Sep):1829–1859.