What do character-level models learn about morphology? The case of dependency parsing

Citation for published version:
Vania, C, Grivas, A & Lopez, A 2018, What do character-level models learn about morphology? The case of dependency parsing. in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium , pp. 2573-2583, 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31/10/18.

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing

What do character-level models learn about morphology? The case of dependency parsing

Clara Vania **Andreas Grivas**[*] **Adam Lopez**
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
c.vania@ed.ac.uk, andreasgrv@gmail.com, alopez@inf.ed.ac.uk

Abstract

When parsing morphologically-rich languages with neural models, it is beneficial to model input at the character level, and it has been claimed that this is because character-level models learn morphology. We test these claims by comparing character-level models to an oracle with access to explicit morphological analysis on twelve languages with varying morphological typologies. Our results highlight many strengths of character-level models, but also show that they are poor at disambiguating some words, particularly in the face of case syncretism. We then demonstrate that explicitly modeling morphological case improves our best model, showing that character-level models can benefit from targeted forms of explicit morphological modeling.

1 Introduction

Modeling language input at the character level (Ling et al., 2015; Kim et al., 2016) is effective for many NLP tasks, and often produces better results than modeling at the word level. For parsing, Ballesteros et al. (2015) have shown that character-level input modeling is highly effective on morphologically-rich languages, and the three best systems on the 45 languages of the CoNLL 2017 shared task on universal dependency parsing all use character-level models (Dozat et al., 2017; Shi et al., 2017; Björkelund et al., 2017; Zeman et al., 2017), showing that they are effective across many typologies.

The effectiveness of character-level models in morphologically-rich languages has raised a question and indeed debate about explicit modeling of morphology in NLP. Ling et al. (2015) propose that "prior information regarding morphology ... among others, should be incorporated" into character-level models, while Chung et al. (2016) counter that it is "unnecessary to consider these prior information" when modeling characters. Whether we need to explicitly model morphology is a question whose answer has a real cost: as Ballesteros et al. (2015) note, morphological annotation is expensive, and this expense could be reinvested elsewhere if the predictive aspects of morphology are learnable from strings.

Do character-level models learn morphology? We view this as an empirical claim requiring empirical evidence. The claim has been tested implicitly by comparing character-level models to word lookup models (Qian et al., 2016; Belinkov et al., 2017). In this paper, we test it explicitly, asking how character-level models compare with an oracle model with access to morphological annotations. This extends experiments showing that character-aware language models in Czech and Russian benefit substantially from oracle morphology (Vania and Lopez, 2017), but here we focus on dependency parsing (§2)—a task that benefits substantially from morphological knowledge—and we experiment with twelve languages using a variety of techniques to probe our models.

Our summary finding is that character-level models lag the oracle in nearly all languages (§3). The difference is small, but suggests that there is value in modeling morphology. When we tease apart the results by part of speech and dependency type, we trace the difference back to the character-level model's inability to disambiguate words even when encoded with arbitrary context (§4). Specifically, it struggles with *case syncretism*, in which noun case—and thus syntactic function—is ambiguous. We show that the oracle relies on morphological case, and that a character-level model provided *only* with morphological case rivals the oracle, even when case is provided by another predictive model (§5). Finally, we show that the crucial morphological features vary by language (§6).

[*]Work done while at the University of Edinburgh.

2 Dependency parsing model

We use a neural graph-based dependency parser combining elements of two recent models (Kiperwasser and Goldberg, 2016; Zhang et al., 2017). Let $w = w_1, \ldots, w_{|w|}$ be an input sentence of length $|w|$ and let w_0 denote an artificial ROOT token. We represent the ith input token w_i by concatenating its *word representation* (§2.3), $\mathbf{e}(w_i)$ and part-of-speech (POS) representation, $\mathbf{p}_i$.[1] Using a semicolon $(;)$ to denote vector concatenation, we have:

$$\mathbf{x}_i = [\mathbf{e}(w_i); \mathbf{p}_i] \tag{1}$$

We call $\mathbf{x}_i$ the *embedding* of w_i since it depends on context-independent word and POS representations. We obtain a context-sensitive *encoding* $\mathbf{h}_i$ with a bidirectional LSTM (bi-LSTM), which concatenates the hidden states of a forward and backward LSTM at position i. Using $\mathbf{h}_i^f$ and $\mathbf{h}_i^b$ respectively to denote these hidden states, we have:

$$\mathbf{h}_i = [\mathbf{h}_i^f; \mathbf{h}_i^b] \tag{2}$$

We use $\mathbf{h}_i$ as the final input representation of w_i.

2.1 Head prediction

For each word w_i, we compute a distribution over all other word positions $j \in \{0, ..., |w|\}/i$ denoting the probability that w_j is the headword of w_i.

$$P_{head}(w_j \mid w_i, w) = \frac{\exp(a(\mathbf{h}_i, \mathbf{h}_j))}{\sum_{j'=0}^{|w|} \exp(a(\mathbf{h}_i, \mathbf{h}_{j'}))} \tag{3}$$

Here, a is a neural network that computes an association between w_i and w_j using model parameters $\mathbf{U}_a$, $\mathbf{W}_a$, and $\mathbf{v}_a$.

$$a(\mathbf{h}_i, \mathbf{h}_j) = \mathbf{v}_a \tanh(\mathbf{U}_a \mathbf{h}_i + \mathbf{W}_a \mathbf{h}_j) \tag{4}$$

2.2 Label prediction

Given a head prediction for word w_i, we predict its syntactic label $\ell_k \in L$ using a similar network.

$$P_{label}(\ell_k \mid w_i, w_j, w) = \frac{\exp(f(\mathbf{h}_i, \mathbf{h}_j)[k])}{\sum_{k'=1}^{|L|} \exp(f(\mathbf{h}_i, \mathbf{h}_j)[k'])} \tag{5}$$

where L is the set of output labels and f is a function that computes label score using model parameters $\mathbf{U}_\ell$, $\mathbf{W}_\ell$, and $\mathbf{V}_\ell$:

$$f(\mathbf{h}_i, \mathbf{h}_j) = \mathbf{V}_\ell \tanh(\mathbf{U}_\ell \mathbf{h}_i + \mathbf{W}_\ell \mathbf{h}_j) \tag{6}$$

The model is trained to minimize the summed cross-entropy losses of both head and label prediction. At test time, we use the Chu-Liu-Edmonds (Chu and Liu, 1965; Edmonds, 1967) algorithm to ensure well-formed, possibly non-projective trees.

2.3 Computing word representations

We consider several ways to compute the word representation $\mathbf{e}(w_i)$ in Eq. 1:

word. Every word type has its own learned vector representation.

char-lstm. Characters are composed using a bi-LSTM (Ling et al., 2015), and the final states of the forward and backward LSTMs are concatenated to yield the word representation.

char-cnn. Characters are composed using a convolutional neural network (Kim et al., 2016).

trigram-lstm. Character trigrams are composed using a bi-LSTM, an approach that we previously found to be effective across typologies (Vania and Lopez, 2017).

oracle. We treat the morphemes of a morphological annotation as a sequence and compose them using a bi-LSTM. We only use universal inflectional features defined in the UD annotation guidelines. For example, the morphological annotation of "chases" is ⟨`chase, person=3rd, num-SG, tense=Pres`⟩.

For the remainder of the paper, we use the name of model as shorthand for the dependency parser that uses that model as input (Eq. 1).

3 Experiments

Data We experiment on twelve languages with varying morphological typologies (Table 1) in the Universal Dependencies (UD) treebanks version 2.0 (Nivre et al., 2017).[2] Note that while Arabic and Hebrew follow a root & pattern typology, their datasets are unvocalized, which might reduce the observed effects of this typology. Following common practice, we remove language-specific dependency relations and multiword token annotations. We use gold sentence segmentation, tokenization, universal POS (UPOS), and morphological (XFEATS) annotations provided in UD.

Implementation and training Our Chainer (Tokui et al., 2015) implementation encodes words (Eq. 2) in two-layer bi-LSTMs with 200 hidden units, and uses 100 hidden units for head and label

[1]This combination yields the best labeled accuracy according to Ballesteros et al. (2015).

[2]For Russian we use the UD_Russian_SynTagRus treebank, and for all other languages we use the default treebank.

Languages	#sents (K)	#tokens (K)	type/token ratio (%)
Finnish	12.2	162.6	28.5
Turkish	3.7	38.1	33.6
Czech	68.5	1173.3	9.5
English	12.5	204.6	8.1
German	14.1	269.6	17.7
Hindi	13.3	281.1	6
Portuguese	8.3	206.7	11.7
Russian	48.8	870	11.4
Spanish	14.2	382.4	11.1
Urdu	4.0	108.7	8.8
Arabic	6.1	223.9	10.3
Hebrew	5.2	137.7	11.7

Table 1: Training data statistics. Languages are grouped by their dominant morphological processes, from top to bottom: agglutinative, fusional, and root & pattern.

predictions (output of Eqs. 4 and 6). We set batch size to 16 for char-cnn and 32 for other models following a grid search. We apply dropout to the embeddings (Eq. 1) and the input of the head prediction. We use Adam optimizer with initial learning rate 0.001 and clip gradients to 5, and train all models for 50 epochs with early stopping. For the word model, we limit our vocabulary to the 20K most frequent words, replacing less frequent words with an unknown word token. The char-lstm, trigram-lstm, and oracle models use a one-layer bi-LSTM with 200 hidden units to compose subwords. For char-cnn, we use the small model setup of Kim et al. (2016).

Parsing Results Table 2 presents test results for every model on every language, establishing three results. First, they support previous findings that character-level models outperform word-based models—indeed, the char-lstm model outperforms the word model on LAS for all languages except Hindi and Urdu for which the results are identical.[3] Second, they establish strong baselines for the character-level models: the char-lstm generally obtains the best parsing accuracy, closely followed by char-cnn. Third, they demonstrate that character-level models rarely match the accuracy of an oracle model with access to explicit morphology. This reinforces a finding of Vania and Lopez (2017): character-level models are effective tools, but they do not learn everything about morphology, and they seem to be closer to oracle accuracy in agglutinative rather than in fusional languages.

[3]Note that Hindi and Urdu are mutually intelligible.

4 Analysis

4.1 Why do characters beat words?

In character-level models, orthographically similar words share many parameters, so we would expect these models to produce good representations of OOV words that are morphological variants of training words. Does this effect explain why they are better than word-level models?

Sharing parameters helps with both seen and unseen words Table 3 shows how the character model improves over the word model for both non-OOV and OOV words. On the agglutinative languages Finnish and Turkish, where the OOV rates are 23% and 24% respectively, we see the highest LAS improvements, and we see especially large improvements in accuracy of OOV words. However, the effects are more mixed in other languages, even with relatively high OOV rates. In particular, languages with rich morphology like Czech, Russian, and (unvocalised) Arabic see more improvement than languages with moderately rich morphology and high OOV rates like Portuguese or Spanish. This pattern suggests that parameter sharing between pairs of *observed* training words can also improve parsing performance. For example, if "dog" and "dogs" are observed in the training data, they will share activations in their context *and* on their common prefix.

4.2 Why do morphemes beat characters?

Let's turn to our main question: what do character-level models learn about morphology? To answer it, we compare the oracle model to char-lstm, our best character-level model.

Morphological analysis disambiguates words In the oracle, morphological annotations disambiguate some words that the char-lstm must disambiguate from context. Consider these Russian sentences from Baerman et al. (2005):

(1) Maša čitaet <u>pis′mo</u>
Masha reads <u>letter</u>
'*Masha reads a <u>letter</u>.*'

Model →	word		char-lstm		char-cnn		trigram-lstm		oracle		o/c
↓ Language	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	LAS
Finnish	85.7	80.8	**90.6**	**88.4**	89.9	87.5	89.7	87.0	90.6	88.8	+0.4
Turkish	71.4	61.6	**74.7**	**68.6**	74.4	67.9	73.2	65.9	75.3	69.5	+0.9
Czech	92.6	89.3	**93.5**	**90.6**	**93.5**	**90.6**	92.7	89.2	94.3	92.0	+1.4
English	90.6	88.9	91.3	89.4	**91.7**	**90.0**	90.4	88.5	91.7	89.9	+0.5
German	**88.1**	**84.5**	88.0	**84.5**	87.8	84.4	87.1	83.5	88.8	86.5	+2.0
Hindi	**95.8**	93.1	95.7	**93.3**	95.7	93.2	93.4	89.8	95.9	93.3	-
Portuguese	87.4	85.5	**87.8**	**86.0**	87.7	**86.0**	86.7	84.8	88.0	86.5	+0.5
Russian	92.4	90.1	**94.0**	**92.4**	93.8	92.1	92.0	89.5	94.4	93.3	+0.9
Spanish	89.4	86.9	89.8	**87.4**	**90.0**	87.3	88.6	85.5	90.0	87.7	+0.3
Urdu	91.1	87.0	91.2	87.1	**91.3**	**87.2**	88.6	83.5	90.9	87.0	-0.1
Arabic	75.5	70.9	**76.7**	72.1	76.6	**72.2**	74.6	68.9	76.7	72.7	+0.6
Hebrew	**73.5**	**69.8**	73.4	**69.8**	73.3	**69.8**	71.3	67.1	73.3	70.0	+0.2

Table 2: Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) on **test set**. The best accuracy for each language is highlighted in **bold** for all models, and for all non-oracle models. **o/c:** LAS improvement from char-lstm to oracle.

Language	dev	LAS improvement	
	%OOV	non-OOV	OOV
Finnish	23.0	6.8	17.5
Turkish	24.0	4.6	13.5
Czech	5.8	1.4	3.9
English	6.8	0.7	5.2
German	9.7	0.9	0.7
Hindi	4.3	0.2	0.0
Portuguese	8.1	0.3	1.3
Russian	8.4	2.1	6.9
Spanish	7.0	0.4	0.7
Arabic	8.0	1.2	7.3
Hebrew	9.0	0.2	1.3

Table 3: LAS improvements (char-lstm − word) for non-OOV and OOV words on development set.

(2) Na stole ležit pis′mo
on table lies letter
'*There's a* letter *on the table.*'

Pis′mo ("letter") acts as the subject in (1), and as object in (2). This knowledge is available to the oracle via morphological case: in (1), the case of *pis′mo* is nominative and in (2) it is accusative. Could this explain why the oracle outperforms the character model?

To test this, we look at accuracy for word types

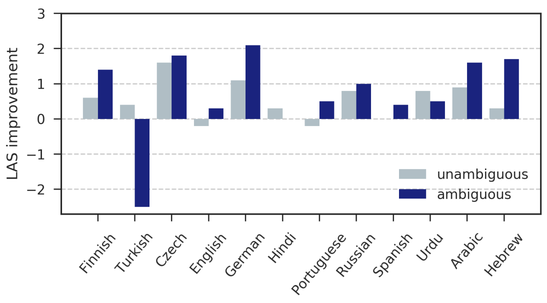

Figure 1: LAS improvements (oracle − char-lstm) for ambiguous and unambiguous words on development set.

that are *empirically* ambiguous—those that have more than one morphological analysis in the training data. Note that by this definition, some ambiguous words will be seen as unambiguous, since they were seen with only one analysis. To make the comparison as fair as possible, we consider only words that were observed in the training data. Figure 1 compares the improvement of the oracle on ambiguous and seen unambiguous words, and as expected we find that handling of ambiguous words improves with the oracle in almost all languages. The only exception is Turkish, which has the least training data.

Morphology helps for nouns Now we turn to a more fine-grained analysis conditioned on the annotated part-of-speech (POS) of the *depen-*

Language	Model	ADJ	NOUN	PRON	PROPN	VERB	Overall
Finnish	%tokens	8.1	32.5	8.2	6.7	16.1	-
	char-lstm	89.2	82.1	88.1	84.5	88.4	87.7
	oracle	90.3	83.3	89.5	86.2	89.3	88.5
	diff	+1.1	+1.2	+1.4	+1.7	+0.9	+0.8
Czech	%tokens	14.9	28.7	3.6	6.3	10.7	-
	char-lstm	94.2	83.6	85.3	84.3	90.7	91.2
	oracle	94.8	87.5	88.5	86.8	91.1	92.5
	diff	+0.6	+3.9	+3.2	+2.5	+0.4	+1.3
German	%tokens	7.6	20.4	9.5	5.6	12.1	-
	char-lstm	88.4	81.4	86.0	82.4	85.2	87.5
	oracle	89.1	87.1	93.2	84.4	86.3	89.7
	diff	+0.7	+5.7	+7.2	+2.0	+1.1	+2.2
Russian	%tokens	12.2	29.3	6.1	4.6	13.7	-
	char-lstm	93.2	86.7	92.0	80.2	88.5	91.6
	oracle	93.7	88.8	93.3	86.4	88.9	92.6
	diff	+0.5	+2.1	+1.3	+6.2	+0.4	+1.0

Table 4: Labeled accuracy for different parts of speech on development set.

dent. We focus on four languages where the oracle strongly outperforms the best character-level model on the development set: Finnish, Czech, German, and Russian.[4] We consider five POS categories that are frequent in all languages and consistently annotated for morphology in our data: adjective (ADJ), noun (NOUN), pronoun (PRON), proper noun (PROPN), and verb (VERB).

Table 4 shows that the three noun categories—ADJ, PRON, and PROPN—benefit substantially from oracle morphology, especially for the three fusional languages: Czech, German, and Russian.

Morphology helps for subjects and objects We analyze results by the dependency type of the *dependent*, focusing on types that interact with morphology: *root*, nominal subjects (*nsubj*), objects (*obj*), indirect objects (*iobj*), nominal modifiers (*nmod*), adjectival modifier (*amod*), obliques (*obl*), and (syntactic) case markings (*case*).

Figure 2 shows the differences in the confusion matrices of the char-lstm and oracle for those words on which both models correctly predict the head. The differences on Finnish are small, which we expect from the similar overall LAS of both models. But for the fusional languages, a pattern emerges: the char-lstm consistently underperforms the oracle on nominal subject, object, and indirect object dependencies—labels closely associated with noun categories. From inspection, it appears to frequently mislabel objects as nominal subjects when the dependent noun is morphologically ambiguous. For example, in the sentence of Figure 3, *Gelände* ("terrain") is an object, but the char-lstm incorrectly predicts that it is a nominal subject. In the training data, *Gelände* is ambiguous: it can be accusative, nominative, or dative.

In German, the char-lstm frequently confuses objects and indirect objects. By inspection, we found 21 mislabeled cases, where 20 of them would likely be correct if the model had access to morphological case (usually dative). In Czech and Russian, the results are more varied: indirect objects are frequently mislabeled as objects, obliques, nominal modifiers, and nominal subjects. We note that indirect objects are relatively rare in these data, which may partly explain their frequent mislabeling.

5 Characters and case syncretism

So far, we've seen that for our three fusional languages—German, Czech, and Russian—the or-

[4]This is slightly different than on the test set, where the effect was stronger in Turkish than in Finnish. In general, we found it difficult to draw conclusions from Turkish, possibly due to the small size of the data.

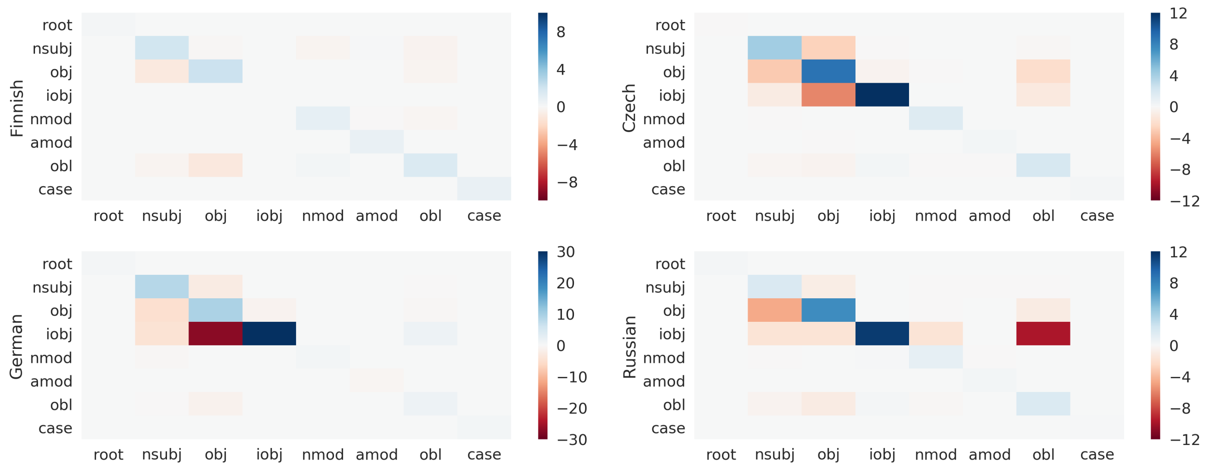

Figure 2: Heatmaps of the difference between oracle vs. char-lstm confusion matrices for label prediction when both head predictions are correct (**x-axis**: predicted labels; **y-axis**: gold labels). Blue cells have higher oracle values, red cells have higher char-lstm values.

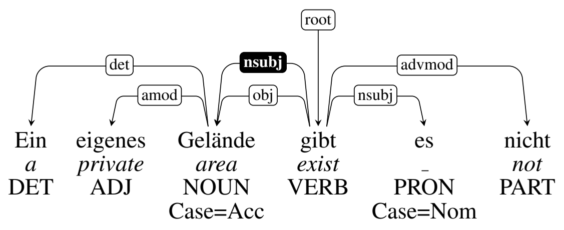

Figure 3: A sentence which the oracle parses perfectly (shown in white) and the char-lstm predicts an incorrect label (shown in black).

acle strongly outperforms a character model on nouns with ambiguous morphological analyses, particularly on core dependencies: nominal subjects, objects and indirect objects. Since the nominative, accusative, and dative morphological cases are strongly (though not perfectly) correlated with these dependencies, it is easy to see why the morphologically-aware oracle is able to predict them so well. We hypothesized that these cases are more challenging for the character model because these languages feature a high degree of *syncretism*—functionally distinct words that have the same form—and in particular case syncretism. For example, referring back to examples (1) and (2), the character model must disambiguate *pis´mo* from its context, whereas the oracle can directly disambiguate it from a feature of the word itself.[5]

To understand this, we first designed an experiment to see whether the char-lstm could successfully disambiguate *noun* case, using a method similar to (Belinkov et al., 2017). We train a neural classifier that takes as input a word representation from the trained parser and predicts a morphological feature of that word—for example that its case is nominative (`Case=Nom`). The classifier is a feedforward neural network with one hidden layer, followed by a ReLU non-linearity. We consider two representations of each word: its embedding ($\mathbf{x}_i$; Eq. 1) and its encoding ($\mathbf{h}_i$; Eq. 2). To understand the importance of case, we consider it alongside number and gender features as well as whole feature bundles.

The oracle relies on case Table 5 shows the results of morphological feature classification on Czech; we found very similar results in German and Russian (Appendix A.2). The oracle embeddings have almost perfect accuracy—and this is just what we expect, since the representation only needs to preserve information from its input. The char-lstm embeddings perform well on number and gender, but less well on case. This results suggest that the character-level models still struggle to learn case when given only the input text. Comparing the char-lstm with a baseline model which predicts the most frequent feature for each type in the training data, we observe that both of them show similar trends even though character models slightly outperforms the baseline model.

The classification results from the encoding are particularly interesting: the oracle still performs very well on morphological case, but less well on other features, even though they appear in

[5]We are far from first to observe that morphological case is important to parsing: Seeker and Kuhn (2013) observe the same for non-neural parsers.

Feature	baseline	embedding		encoder	
		char	oracle	char	oracle
Case	71.1	74.4	**100**	86.5	**98.6**
Gender	92.9	98.1	**100**	**71.2**	58.6
Number	88.9	94.7	**100**	84.2	**84.8**
All	70.4	72.5	**99.9**	**58.1**	50.2

Table 5: Morphological tagging accuracy from representations using the char-lstm and oracle embedding and encoder representations in Czech. Baseline simply chooses the most frequent tag. *All* means we concatenate all annotated features in UD as one tag.

the input. In the character model, the accuracy in morphological prediction also degrades in the encoding—*except* for case, where accuracy on case improves by 12%.

These results make intuitive sense: representations learn to preserve information from their input that is useful for subsequent predictions. In our parsing model, morphological case is very useful for predicting dependency labels, and since it is present in the oracle's input, it is passed almost completely intact through each representation layer. The character model, which must disambiguate case from context, draws as much additional information as it can from surrounding words through the LSTM encoder. But other features, and particularly whole feature bundles, are presumably less useful for parsing, so neither model preserves them with the same fidelity.[6]

Explicitly modeling case improves parsing accuracy Our analysis indicates that case is important for parsing, so it is natural to ask: Can we improve the neural model by explicitly modeling case? To answer this question, we ran a set of experiments, considering two ways to augment the char-lstm with case information: multi-task learning (MTL; Caruana, 1997) and a pipeline model in which we augment the char-lstm model with either predicted or gold case. For example, we use ⟨`p, i, z, z, a, Nom`⟩ to represent *pizza* with nominative case. For MTL, we follow the setup of Søgaard and Goldberg (2016) and Coavoux and Crabbé (2017). We increase the biLSTMs layers from two to four and use the first two layers to predict morphological case, leaving out the other two layers specific only for parser. For the pipeline model, we train a morphological tagger to predict morphological case (Appendix A.1). This tagger does not share parameters with the parser.

[6]This finding is consistent with Ballesteros (2013) which performed careful feature analysis on morphologically rich languages and found that lemma and case features provide the highest improvement in a non-neural transition based parser compared to other features.

Language	Input	Dev	Test
Czech	char	91.2	90.6
	char (multi-task)	91.6	91.0
	char + predicted case	**92.2**	**91.8**
	char + gold case	92.3	91.9
	oracle	92.5	92.0
German	char	87.5	84.5
	char (multi-task)	**87.9**	84.4
	char + predicted case	87.8	**86.4**
	char + gold case	90.2	86.9
	oracle	89.7	86.5
Russian	char	91.6	92.4
	char (multi-task)	92.2	92.6
	char + predicted case	**92.5**	**93.3**
	char + gold case	92.8	93.5
	oracle	92.6	93.3

Table 6: LAS results when case information is added. We use **bold** to highlight the best results for models without explicit access to gold annotations.

Table 6 summarizes the results on Czech, German, and Russian. We find augmenting the char-lstm model with either oracle or predicted case improve its accuracy, although the effect is different across languages. The improvements from predicted case results are interesting, since in non-neural parsers, predicted case usually harms accuracy (Tsarfaty et al., 2010). However, we note that our taggers use gold POS, which might help. The MTL models achieve similar or slightly better performance than the character-only models, suggesting that supplying case in this way is beneficial. Curiously, the MTL parser is worse than the the pipeline parser, but the MTL case tagger is better than the pipeline case tagger (Table 7). This indicates that the MTL model must learn to encode case in the model's representation, but must not learn to effectively use it for parsing. Finally, we

Language	%case	Dev		Test	
		PL	MT	PL	MT
Czech	66.5	95.4	**96.7**	95.2	**96.6**
German	36.2	**92.6**	92.0	90.8	**91.4**
Russian	55.8	95.8	**96.5**	95.9	**96.5**

Table 7: Case accuracy for case-annotated tokens, for pipeline (**PL**) vs. multitask (**MT**) setup. **%case** shows percentage of training tokens annotated with case.

observe that augmenting the char-lstm with either gold or predicted case improves the parsing performance for all languages, and indeed closes the performance gap with the full oracle, which has access to *all* morphological features. This is especially interesting, because it shows using carefully targeted linguistic analyses can improve accuracy as much as wholesale linguistic analysis.

6 Understanding head selection

The previous experiments condition their analysis on the *dependent*, but dependency is a relationship between dependents and heads. We also want to understand the importance of morphological features to the *head*. Which morphological features of the head are important to the oracle?

Composing features in the oracle To see which morphological features the oracle depends on when making predictions, we augmented our model with a **gated attention mechanism** following Kuncoro et al. (2017). Our new model attends to the morphological features of candidate head w_j when computing its association with dependent w_i (Eq. 3), and morpheme representations are then scaled by their *attention weights* to produce a final representation.

Let $f_{i1}, \cdots, f_{ik}$ be the k morphological features of w_i, and denote by $\mathbf{f}_{i1}, \cdots, \mathbf{f}_{ik}$ their corresponding *feature embeddings*. As in §2, $\mathbf{h}_i$ and $\mathbf{h}_j$ are the encodings of w_i and w_j, respectively. The morphological representation $\mathbf{m}_j$ of w_j is:

$$\mathbf{m}_j = [\mathbf{f}_{j1}, \cdots, \mathbf{f}_{jk}]^\top \mathbf{k} \tag{7}$$

where $\mathbf{k}$ is a vector of attention weights:

$$\mathbf{k} = \text{softmax}([\mathbf{f}_{j1}, \cdots, \mathbf{f}_{jk}]^\top \mathbf{V}\mathbf{h}_i) \tag{8}$$

The intuition is that dependent w_i can choose which morphological features of w_j are most important when deciding whether w_j is its head. Note that this model is asymmetric: a word only attends to the morphological features of its (single) parent, and not its (many) children, which may have different functions. [7]

We combine the morphological representation with the word's encoding via a sigmoid gating mechanism.

$$\mathbf{z}_j = \mathbf{g} \odot \mathbf{h}_j + (1 - \mathbf{g}) \odot \mathbf{m}_j \tag{9}$$

$$\mathbf{g} = \sigma(\mathbf{W}_1 \mathbf{h}_j + \mathbf{W}_2 \mathbf{m}_j) \tag{10}$$

where $\odot$ denotes element-wise multiplication. The gating mechanism allows the model to choose between the computed word representation and the weighted morphological representations, since for some dependencies, morphological features of the head might not be important. In the final model, we replace Eq. 3 and Eq. 4 with the following:

$$P_{head}(w_j|w_i, w) = \frac{\exp(a(\mathbf{h}_i, \mathbf{z}_j))}{\sum_{j'=0}^{N} \exp a(\mathbf{h}_i, \mathbf{z}_{j'})} \tag{11}$$

$$a(\mathbf{h}_i, \mathbf{z}_j) = \mathbf{v}_a \tanh(\mathbf{U}_a \mathbf{h}_i + \mathbf{W}_a \mathbf{z}_j) \tag{12}$$

The modified label prediction is:

$$P_{label}(\ell_k|w_i, w_j, w) = \frac{\exp(f(\mathbf{h}_i, \mathbf{z}_j)[k])}{\sum_{k'=0}^{|L|} \exp(f(\mathbf{h}_i, \mathbf{z}_j)[k'])} \tag{13}$$

where f is again a function to compute label score:

$$f(\mathbf{h}_i, \mathbf{z}_j) = \mathbf{V}_\ell \tanh(\mathbf{U}_\ell \mathbf{h}_i + \mathbf{W}_\ell \mathbf{z}_j) \tag{14}$$

Attention to headword morphological features We trained our augmented model (**oracle-attn**) on Finnish, German, Czech, and Russian. Its accuracy is very similar to the oracle model (Table 8), so we obtain a more interpretable model with no change to our main results.

Next, we look at the learned attention vectors to understand which morphological features are important, focusing on the core arguments: nominal subjects, objects, and indirect objects. Since our model knows the case of each dependent, this enables us to understand what features it seeks in potential heads for each case. For simplicity, we only report results for words where both head and label predictions are correct.

[7]This is a simple and much less computationally demanding variant of the model of Dozat et al. (2017), which uses different views for each head/dependent role.

Language	oracle		oracle-attn	
	UAS	LAS	UAS	LAS
Finnish	**89.2**	**87.3**	88.9	86.9
Czech	93.4	91.3	**93.5**	**91.3**
German	90.4	88.7	**90.7**	**89.1**
Russian	**93.9**	**92.8**	93.8	92.7

Table 8: Our attention experiment results on development set.

Figure 4 shows how attention is distributed across multiple features of the head word. In Czech and Russian, we observe that the model attends to *Gender* and *Number* when the noun is in nominative case. This makes intuitive sense since these features often signal subject-verb agreement. As we saw in earlier experiments, these are features for which a character model can learn reliably good representations. For most other dependencies (and all dependencies in German), *Lemma* is the most important feature, suggesting a strong reliance on lexical semantics of nouns and verbs. However, we also notice that the model sometimes attends to features like *Aspect*, *Polarity*, and *VerbForm*—since these features are present only on verbs, we suspect that the model may simply use them as convenient signals that a word is verb, and thus a likely head for a given noun.

7 Conclusion

Character-level models are effective because they can represent OOV words and orthographic regularities of words that are consistent with morphology. But they depend on context to disambiguate words, and for some words this context is insufficient. Case syncretism is a specific example that our analysis identified, but the main results in Table 2 hint at the possibility that different phenomena are at play in different languages.

While our results show that prior knowledge of morphology is important, they also show that it can be used in a targeted way: our character-level models improved markedly when we augmented them only with case. This suggests a pragmatic reality in the middle of the wide spectrum between pure machine learning from raw text input and linguistically-intensive modeling: our new models don't need all prior linguistic knowledge, but they clearly benefit from some knowledge in addition to raw input. While we used a data-driven analysis to identify case syncretism as a problem for neural parsers, this result is consistent with previous linguistically-informed analyses (Seeker and Kuhn, 2013; Tsarfaty et al., 2010). We conclude that neural models can still benefit from linguistic analyses that target specific phenomena where annotation is likely to be useful.

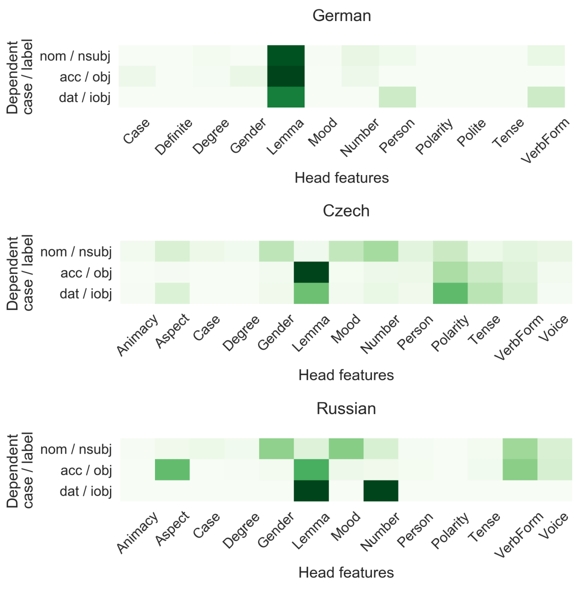

Figure 4: The importance of morphological features of the head for subject and object predictions.

Acknowledgments

Clara Vania is supported by the Indonesian Endowment Fund for Education (LPDP), the Centre for Doctoral Training in Data Science, funded by the UK EPSRC (grant EP/L016427/1), and the University of Edinburgh. We would like to thank Yonatan Belinkov for the helpful discussion regarding morphological tagging experiments. We thank Sameer Bansal, Marco Damonte, Denis Emelin, Federico Fancellu, Sorcha Gilroy, Jonathan Mallinson, Joana Ribeiro, Naomi Saphra, Ida Szubert, Sabine Weber, and the anonymous reviewers for helpful discussion of this work and comments on previous drafts of the paper.

References

Matthew Baerman, Dunstan Brown, and Greville G. Corbett. 2005. *The Syntax-Morphology Interface: A Study of Syncretism*. Cambridge Studies in Linguistics. Cambridge University Press.

Miguel Ballesteros. 2013. Effective morphological feature selection with MaltOptimizer at the

SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 63–70, Seattle, Washington, USA. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Anders Björkelund, Agnieszka Falenska, Xiang Yu, and Jonas Kuhn. 2017. IMS at the CoNLL 2017 UD shared task: CRFs and perceptrons meet neural networks. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 40–51, Vancouver, Canada. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Y. J. Chu and T. H. Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Maximin Coavoux and Benoit Crabbé. 2017. Multilingual lexicalized constituency parsing with word-level auxiliary tasks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 331–336. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-aware neural language models. In *Proceedings of the 2016 Conference on Artificial Intelligence (AAAI)*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What do recurrent neural network grammars learn about syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Joakim Nivre et al. 2017. Universal Dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague, http://hdl.handle.net/11234/1-1983.

Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. Investigating language universal and specific properties in word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Tianze Shi, Felix G. Wu, Xilun Chen, and Yao Cheng. 2017. Combining global models for parsing universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 31–39, Vancouver, Canada. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235. Association for Computational Linguistics.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings* *of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

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*, SPMRL '10, pages 1–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2016–2027. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağr Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 665–676, Valencia, Spain. Association for Computational Linguistics.

A Supplemental Material

A.1 Morphological tagger

We adapt the parser's encoder architecture for our morphological tagger. Following notation in Section 2, each word w_i is represented by its context-sensitive encoding, $\mathbf{h}_i$ (Eq. 2). The encodings are then fed into a feed-forward neural network with two hidden layers—each has a ReLU non-linearity—and an output layer mapping the to the morphological tags, followed by a softmax. We set the size of the hidden layer to 100 and use dropout probability 0.2. We use Adam optimizer with initial learning rate 0.001 and clip gradients to 5. We train each model for 20 epochs with early stopping. The model is trained to minimized the cross-entropy loss.

Since we do not have additional data with the same annotations, we use the same UD dataset to train our tagger. To prevent overfitting, we only use the first 75% of training data for training[8]. After training the taggers, we predict the case for the training, development, and test sets and use them for dependency parsing.

A.2 Results on morphological tagging

Table 9 and 10 present morphological tagging results for German and Russian. We found that German and Russian have similar pattern to Czech (Table 5), where morphological case seems to be preserved in the encoder because they are useful for dependency parsing. In these three fusional languages, contextual information helps character-level model to predict the correct case. However, its performance still behind the oracle.

We observe a slightly different pattern on Finnish results (Table 11). The character embeddings achieves almost similar performance as the oracle embeddings. This results highlights the differences in morphological process between Finnish and the other fusional languages. We observe that performance of the encoder representations are slightly worse than the embeddings.

[8]We tried other settings, i.e. 25%, 50%, 100%, but in general we achieve best result when we use 75% of the training data.

Feature	baseline	embedding		encoder	
		char	oracle	char	oracle
Case	35.2	35.7	**100**	80.8	**99.7**
Gender	56.8	63.6	**100**	75.7	**78**
Number	59.1	67.1	**100**	78.3	**93.9**
All	34	34.3	**100**	63.6	**78.5**

Table 9: Morphological tagging results for German.

Feature	baseline	embedding		encoder	
		char	oracle	char	oracle
Case	71.6	80.5	**100**	90.4	**98.5**
Gender	87.7	97.4	**100**	**69.9**	57.3
Number	83.7	94.5	**100**	**85.7**	83.8
All	71.3	77.2	**99.9**	**56.9**	47.2

Table 10: Morphological tagging results for Russian.

Feature	baseline	embedding		encoder	
		char	oracle	char	oracle
Case	56	96.7	**100**	88.9	**91.4**
Number	56.4	97.4	**100**	81.9	**89.5**
All	55.8	**95**	91.6	74	**82.7**

Table 11: Morphological tagging results for Finnish.