Character Eyes: Seeing Language through Character-Level Taggers

Yuval Pinter **Marc Marone** **Jacob Eisenstein**
Georgia Institute of Technology
{uvp, mmarone6, jacobe}@gatech.edu

Abstract

Character-level models have been used extensively in recent years in NLP tasks as both supplements and replacements for closed-vocabulary token-level word representations. In one popular architecture, character-level RNNs, typically LSTMs, form a bottom tier creating a word representation for a sequence tagger used to predict token-level annotations such as part-of-speech (POS) tags.

In this work, we examine the behavior of POS taggers from the perspective of individual hidden units within the character-level LSTM. Analysis of activation patterns on a macro scale allows us to identify the ways in which the burden of POS detection is spread across the hidden layer in different languages, as a function of their morphological properties. Using ablation tests, we show how different allocations of forward and backward units affect model arrangement and performance in different categories of languages. We use these results to offer heuristics for hyperparameter selection that are based on known linguistic traits.

1 Introduction

Neural models that learn representations for entities below the word level are becoming a standard in natural language processing tasks (e.g., Bojanowski et al., 2017; Peters et al., 2018). In particular, character representations have been shown to handle out-of-vocabulary words in supervised tagging tasks (Ling et al., 2015; Lample et al., 2016). These advantages generalize across multiple languages, where morphological formation may differ greatly but the character composition of words remains a relatively reliable primitive (Plank et al., 2016).

While the advantages of character-level models are readily apparent, less is understood about the exact mechanism by which these models encode linguistic knowledge about morphology and orthography. Different languages exhibit character-word correspondence in very different patterns, and yet the bi-directional LSTM appears to be, or is assumed to be, capable of capturing them all. In large multilingual settings, it is not uncommon to tune hyperparameters on a handful of languages, and apply them to the rest (e.g., Pinter et al., 2017).

In this work, we challenge this implicit generalization. We train character-based sequence taggers on a large selection of languages exhibiting various strategies for word formation, and subject the resulting models to a novel analysis of the behavior of individual units in the character-level Bi-LSTM hidden layer. This reveals differences in the ability of the Bi-LSTM architecture to identify parts-of-speech, based on typological properties: hidden layers trained on agglutinative languages find more regularities on the character level than in fusional languages; languages that are suffix-heavy give a stronger signal to the backward-facing hidden units, and vice versa for prefix-heavy languages. In short, character-level recurrent networks function differently depending on how each language expresses morphosyntactic properties in characters.

These empirical results motivate a novel Bi-LSTM architecture, in which the number of hidden units is unbalanced across the forward and backward directions. We find empirical correspondence between the analytical findings above and performance of such unbalanced Bi-LSTM models, allowing us to translate the typological properties of a language into concrete recommendations for model selection.[1]

[1] All of our code will be made available upon publication.

2 Related Work

Several recent papers attempt to explain neural network performance by investigating hidden state activation patterns on auxiliary or downstream tasks. On the word level, Linzen et al. (2016) trained LSTM language models, evaluated their performance on grammatical agreement detection, and analyzed activation patterns within specific hidden units. We build on this analysis strategy as we aggregate (character-) sequence activation patterns across all hidden units in a model into quantitative measures.

Substantial prior work exists on the character level as well (Karpathy et al., 2015; Vania and Lopez, 2017; Kementchedjhieva and Lopez, 2018; Gerz et al., 2018). Smith et al. (2018) examined the character component in multilingual parsing models empirically, comparing it to the contribution of POS embeddings and pre-trained embeddings. Chaudhary et al. (2018) leveraged cross-lingual character-level correspondence to train NER models for low-resource languages. Most related to our work is Godin et al. (2018), who compared CNN and LSTM character models on a type-level prediction task on three languages, using the post-network softmax values to see which models identify useful character sequences. Unlike their analysis, we examine a more applied token-level task (POS tagging), and focus on the hidden states within the LSTM model in order to analyze its raw view of word composition.

Our initial analysis targeted the characterization of **unit roles**, where a single hidden unit is observed to have some specific function. Findings from e.g. Linzen et al. (2016) suggest that a single hidden unit can learn to track complex syntactic rules. Radford et al. (2017) find that a character level language model can learn a single unit that tracks sentiment without being directly supervised. (Kementchedjhieva and Lopez, 2018) also examine individual units in a character model and find complex behavior by inspecting activation patterns by hand. In contrast, our initial metrics were motivated by discovering these units automatically.

3 Tagging Task

We train a set of LSTM tagging models, following the setup of Ling et al. (2015). A word representation trained from a character-level LSTM submodule is fed into a word-level bidirectional LSTM, with each word's hidden state subsequently fed into a two-layer perceptron producing tag scores, which are then softmaxed to produce a tagging distribution. For languages with additional morphosyntactic attribute tagging, we follow the architecture in Pinter et al. (2017) where the same word-level Bi-LSTM states are used to predict each attribute's value using its own perceptron+softmax scaffolding. In order to produce informative character models, we do not include word-level embeddings, pre-trained or otherwise, in our setup. Hyperparameters , failed models, and data preparation details are given in Appendix C.

Language	Affix†	Morph	POS Accuracy %	
		synth‡	Dev	Test
Arabic	S	int	96.11	95.93
Bulgarian	S	fus	97.91	97.80
Coptic	p	agg	92.54	92.51
Danish	S	fus	95.59	95.46
Greek	S	fus	96.13	96.46
English	S	fus	93.65	93.30
Spanish	S	fus	95.75	95.00
Basque	=	agg	92.99	92.43
Persian	s	fus	96.07	96.10
Irish	=	fus		89.35
Hebrew	s	int	95.71	94.60
Hindi	S	fus	95.03	94.91
Hungarian	S	agg	94.14	92.00
Indonesian	S	iso	92.55	92.68
Italian	S	fus	96.82	96.95
Latvian	s	fus	94.70	93.09
Russian	S	fus	95.29	95.25
Swedish	S	fus	95.80	95.73
Tamil	S	agg	86.46	87.58
Thai	∅	fus	91.37	
Turkish	S	agg	92.08	92.48
Ukrainian	S	fus	95.68	95.26
Vietnamese	∅	iso	88.51	86.58
Chinese	S	iso	93.05	93.11

Table 1: Attributes and tagging accuracy by language (Irish and Thai do not have both dev and test sets). †Affixation: S/s is strongly/weakly suffixing; P/p is strongly/weakly prefixing; = is equally prefixing/suffixing; ∅ is little affixation. ‡Morphological synthesis: agglutinative, fusional, introflexive, isolating.

3.1 Language Selection

As our goal is to examine the relationship between character-level modeling and linguistic properties, we drove language selection based on two morphological properties deemed relevant to the architectural effects examined. All 24 datasets were obtained from Universal Dependencies (UD) version 2.3 (Nivre et al., 2018), and linguistic properties were found in the World Atlas of Language Structures (Bickel and Nichols, 2013; Dryer, 2013).

The selected languages and their properties are presented in Table 1.

Affixation. To evaluate the role of forward and backward units in a bidirectional model, we selected all languages available in UD which are not classified as either weakly or strongly suffixing in inflectional morphology (the vast majority of UD languages). This includes a single prefixing language (Coptic), two equally suffixing and prefixing languages (Basque and Irish), and two languages with little affixation (Thai and Vietnamese).

Morphological Synthesis. Linguistically functional features vary between being expressed as distinct tokens (isolating languages), detectable unique character substrings (agglutinative), fused together but still distinguishable from the stem (fusional), and non-linearly represented within the word form (introflexive). This property has previously been found to affect performance in character-level models (Pinter et al., 2017; Gerz et al., 2018; Chaudhary et al., 2018), and thus we select representatives of each group.

3.2 Results

In our basic setup, we represent words using a concatenation of the final states from a bidirectional character-level LSTM with 64 forward and backward hidden units each. The results for POS tagging, presented in Table 1, are on par with similar models (Plank et al., 2016, for example) despite not including a word-level type embedding component. We attribute this success to our large character embedding size of 256, corroborating findings reported by Smith et al. (2018).

4 Analysis

We next analyze the models trained on the tagging task in an attempt to see how their character-level hidden states encode different manifestations of linguistic information. [2]

4.1 Metrics

For each language, we run the character-level BiLSTM from the trained tagger on POS-unambiguous word types occurring frequently in the training set, grouped into their parts of speech.[3] On each word w, we observe each hidden unit h_i's activation level (output) on each character h_i^c. We obtain a **base measure** $b(w,i)$ based on the activation pattern. For example, an *average absolute* base measure is defined as the average of absolute value activations:

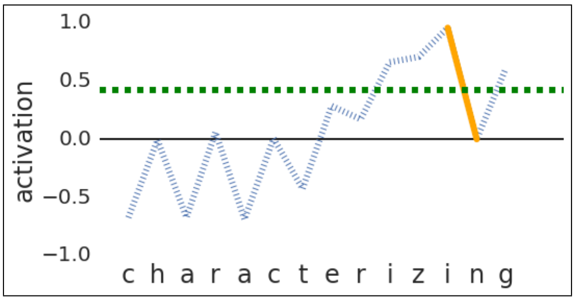

Figure 1: Activations of the English model's unit 42 (forward) on the word *characterizing*. b_{mad} is 0.96 (the drop from the second *i* to *n*), and $b_{\text{avg}|\cdot|}$ is 0.42.

$$b_{\text{avg}|\cdot|}(w,i) = \frac{1}{|w|}\sum_{c=1}^{|w|}|h_i^c|.$$

The *max absolute diff* base measure is defined as:

$$b_{\text{mad}}(w,i) = \max_{c=1}^{|w|-1}|h_i^{c+1} - h_i^c|.$$

Figure 1 demonstrates these two metrics for a sample (word, unit) pair.

Motivated by automatic discovery of unit roles, we developed an initial metric based only on the difference between base metric averages for nouns and verbs, on a particular hidden unit i.

$$\frac{\sum_j b(n_j,i)}{|n|} - \frac{\sum_k b(v_k,i)}{|v|}$$

Where n and v are sets of unambiguous nouns and verbs as defined previously, sampled so $|v|= |n|$. For clarity, we will refer to this as **PPDI**, or Pairwise POS Discrimination Index. Figure 2 shows PPDI values on each of the 128 hidden units in an early English model, using the $b_{\text{avg}|\cdot|}$ metric. Clearly, the last 64 units (the backward units) behave differently. The large spikes indicate units that have a high average value when processing a verb as opposed to a lower value when processing a noun.

[2] Appendix D. contains the details of an additional method involving direct analysis of the weight matrix, developed prior to our focus on hidden states.

[3] We used 8 as our frequency threshold, and define unambiguous forms as ones tagged at least 60% of the time with a given POS.

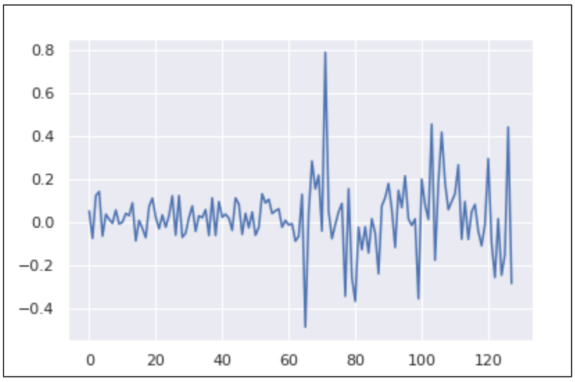

Figure 2: PPDI values for 128 units in an English model, comparing nouns and verbs using the $b_{\text{avg}|\cdot|}$ metric.

PPDI allows for intuitive analysis, but it cannot capture more complicated hidden unit behavior and is limited to pairwise comparisons. To address these issues, we derive a language-level metric for each hidden unit, based on the principle of Mutual Information (MI). The base metric's range ($(0, 1)$ for $b_{\text{avg}|\cdot|}$, $[0, 2)$ for b_{mad}) is divided into B bins of equal size, and base activations from each word are summed across each of the T POS tag categories[4], then normalized to produce a joint probability distribution. The mutual information is computed as:

$$\sum_{t=1}^{T}\sum_{b=1}^{B} P(t, b)[\ln P(t, b) - \ln P(t) - \ln P(b)],$$

and we call the resulting number the POS-Discrimination Index, or **PDI**. Intuitively, a higher PDI implies that the unit activates differently on words of different parts of speech.

At this point a language produces a set of d_h PDIs, one for each unit. We order them from high to low, and define two language-level metrics. The **mass** is the sum of PDI values for all units, $\mathcal{M}(\mathcal{L}) := \sum_{i=1}^{d_h} \text{PDI}(\mathcal{L}, i)$. The **head forwardness** is the proportion of forward-directional units before the point at which half of the mass accumulates (in a random setup, this number would tend to 0.5):

$$\frac{\left|\left\{k : \sum_{i=1}^{k} \text{PDI}(\mathcal{L}, i) \leq \frac{\mathcal{M}(\mathcal{L})}{2} \wedge h_k \text{ is forward}\right\}\right|}{\left|\left\{k : \sum_{i=1}^{k} \text{PDI}(\mathcal{L}, i) \leq \frac{\mathcal{M}(\mathcal{L})}{2}\right\}\right|}$$

These correspond, in theory, to the model's ability to tell POS apart, and to the relative importance of forward and backward units in that role.

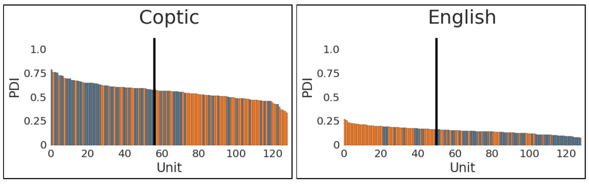

Figure 3: Distribution of PDI values ($b_{\text{avg}|\cdot|}$) across hidden units in Coptic and English, shown in ordered PDI values from largest to smallest, with blue (orange) bars indicating forward (backward) units. The black line demarcates the median point of mass accumulation.

4.2 PDI Patterns

We present the PDI patterns on the $b_{\text{avg}|\cdot|}$ base measure with $B = 16$ bins for Coptic, a prefixing agglutinative language, and English, a suffixing fusional language, in Figure 3.[5] Consistent with other agglutinative languages, Coptic's cumulative mass is very large ($\mathcal{M}(\text{cop}) = 58.1$), suggesting the predictive qualities of the sequence-based LSTM allows good discrimination from the character signal. $\mathcal{M}(\text{eng}) = 16$, demonstrating the difficulty presented by fusional languages. The accumulation of 71% forward (80% backward) units in the head of the Coptic (English) chart suggests an interesting relationship between affixation and LSTM direction: LSTM units are likely to hone in on the POS-indicative signals in the beginning of their run. Unfortunately, since no other prefixing languages are available in UD, we were not able to pursue this hypothesis further. We do note that languages with little or balanced affixation presented directionality statistics between those of Coptic and English.[6]

4.3 Asymmetric Bidirectionality

Using these observations, we conduct a directionality balance study, where we vary the number of hidden units in each of the forward and backwards dimensions. Our analyzed models use 64 forward and 64 backward units (denoted hereafter 64/64), and we thus trained models with imbalanced directionality (128/0, 96/32, 32/96, 0/128). We test the hypothesis that imbalanced models affect languages differently based on their linguistic properties and statistical metrics.

[4] We omit the following 'character-simple' part-of-speech tags: INTJ, NUM, PROPN, PUNCT, SYM, X.

[5] Trends for b_{mad} are similar.

[6] The full scope of this analysis is available in Appendix B.

Language Type	128/0	96/32	64/64 (base)	32/96	0/128
Inflectional Affixation Categories					
S. suffix	**+0.23**	+0.05	94.51	**-0.12**	-0.03
W. suffix	**+0.15**	**+0.20**	95.47	**-0.13**	+0.02
Equal p/s	**+0.50**	**+0.32**	90.92	-0.13	-0.03
Little aff.	-0.08	-0.18	89.81	-0.13	+0.11
W. prefix	**+0.56**	**+0.33**	92.58	**+0.56**	**+0.38**
Morphological Synthesis Categories					
Introflex.	+0.10	+0.10	95.83	+0.01	+0.07
Fusional	**+0.20**	+0.04	94.89	-0.03	**+0.06**
Agglutina.	**+0.54**	**+0.30**	91.72	-0.28	-0.17
Isolating	-0.01	-0.10	91.28	**-0.18**	+0.03
Overall	**+0.23**	**+0.08**	93.86	**-0.10**	+0.01

Table 2: Imbalanced models' mean POS accuracy on UD development data (differences from three random base model runs averaged; **boldfaced** when significant at $p < 0.05$ using a paired two-tailed t-test).

The results for this study are presented in Table 2 as averages for the language categories listed in Table 1. The full results are available in Appendix A. One trend is the preference of **agglutinative** languages for imbalanced models, whereas the other languages are little affected by this change. This could be explained by the increase in inter-unit interaction in the larger direction of an imbalanced model – contiguous character sequences consistently code reliable linguistic features in these languages. A second finding is the slight bias of suffixing languages towards more forward units and of the prefixing language to more backward units, indicating that hidden LSTM units are better in detecting formations close to their final state. We also note the stability of introflexive and little-affixing languages to directionality balance, possibly owing to the relatively small significance of contiguous character sequences in detecting word role. Lastly, we point out that the compromise *sesquidirectional* models 96/32 and 32/96 did not tend to stand out significantly on our tested language categories.

5 Future Work

Future work could investigate the practical implications of our observations. Our work here assumes a fixed combined *forward* + *backward* dimension of 128. However, depending on the implementation of the network, it may be more practical to consider networks with identical parameter counts (an RNN with hidden size 128 has more parameters than 2 RNNs with hidden size 64).

Our experiments involved only a single prefixing language (Coptic). Additional datasets for prefixing languages could reveal further insights. Future work could also consider the creation of synthetic languages to test the extremes of the trends we observe.

Further investigation of unit roles could take several directions. If multiple units are responsible for discriminating between certain parts of speech, redundant units could be pruned to compress the model. Additionally, our PDI metric could be use to create an interpretability measure, as a model where distinct units are responsible for certain morphological properties could be considered more interpretable. In this case, PDI would allow for comparisons independent of accuracy.

6 Conclusion

While character-level Bi-LSTM models compute meaningful word representations across many languages, the way they do it depends on each language's typological properties. These observations can guide model selection: for example, in agglutinative languages we observe a strong preference for a single direction of analysis, motivating the use of unidirectional character-level LSTMs for at least this type of language.

References

Balthasar Bickel and Johanna Nichols. 2013. Fusion of selected inflectional formatives. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.

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.

Matthew S. Dryer. 2013. Prefixing vs. suffixing in inflectional morphology. In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In D. D. Lee, M. Sugiyama,

U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1019–1027. Curran Associates, Inc.

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association of Computational Linguistics*, 6:451–465.

Fréderic Godin, Kris Demuynck, Joni Dambre, Wesley De Neve, and Thomas Demeester. 2018. Explaining character-aware neural networks for word-level prediction: Do they discover linguistic rules? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3275–3284.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Yova Kementchedjhieva and Adam Lopez. 2018. 'indicatements' that character language models learn english morpho-syntactic units and regularities. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 145–153.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W Black, and Isabel Trancoso. 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*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association of Computational Linguistics*, 4(1):521–535.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Joakim Nivre et al. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking Word Embeddings using Subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 102–112.

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)*, pages 412–418. Association for Computational Linguistics.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An investigation of the interactions between pre-trained word embeddings, character models and pos tags in dependency parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2711–2720.

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)*, volume 1, pages 2016–2027.

Language	128/0	96/32	64/64	32/96	0/128
Arabic	96.32	96.11	96.06	96.11	96.14
Bulgarian	98.00	97.84	97.84	97.69	97.74
Coptic	93.14	92.91	92.58	93.14	92.96
Danish	96.07	95.56	95.61	95.56	95.58
Greek	96.13	95.93	96.01	95.96	95.98
English	93.82	93.69	93.65	93.81	93.88
Spanish	95.76	95.56	95.64	95.63	95.81
Basque	93.51	93.10	92.89	92.41	92.93
Persian	96.25	96.27	96.11	95.98	96.13
Irish	89.33	89.39	88.96	89.17	88.86
Hebrew	95.55	95.75	95.60	95.57	95.67
Hindi	95.51	95.14	95.12	95.03	95.40
Hungarian	94.32	94.51	94.20	93.88	93.83
Indonesian	92.48	92.33	92.49	92.52	92.50
Italian	96.98	96.84	96.87	96.82	97.04
Latvian	95.05	94.99	94.69	94.45	94.66
Russian	95.49	95.40	95.32	95.29	95.34
Swedish	95.89	95.49	95.65	95.57	95.58
Tamil	87.25	87.09	86.88	85.75	85.99
Thai	91.22	91.19	91.39	91.46	91.55
Turkish	93.10	92.49	92.06	92.05	92.05
Ukrainian	95.71	95.74	95.63	95.64	95.70
Vietnamese	88.24	88.06	88.23	87.90	88.28
Chinese	93.07	93.15	93.12	92.88	93.15

Table 3: Full balance model accuracy scores. 64/64 numbers are averaged over three random seed runs.

A Full Imbalance Scores

Table 3 presents all scores for balance-different models (experiment in § 4.3).

B Full PDI Scores

Table 4 presents the full numbers for the PDI calculation on all languages.

C Experimental Details

Dataset Selection and Preparation. Most languages have only a single UD 2.3 treebank. For languages with multiple treebanks we selected the largest, except in the cases of Spanish and Indonesian, where we selected the GSD treebanks. The Irish IDT treebank had only a train and test split, so we used the test set for early stopping. The Thai PUD treebank only provided a single dataset with 1000 instances, which we shuffled and partitioned into a 850/150 split. Tokens were normalized to remove noisy data: tokens containing 'http' were replaced with 'URL' and tokens containing '@' were replaced with 'EMAIL'. This was most relevant (293 replacements) for the English treebank, which contained many long URLs.

Language	Mass	Mass median idx	% of forward units until median
Tamil	71.0	55	49.1
Irish	62.0	56	42.9
Coptic	58.1	56	71.4
Hungarian	47.9	55	50.9
Greek	31.2	55	45.5
Turkish	30.1	54	57.4
Russian	25.9	54	40.7
Thai	25.9	55	47.3
Ukrainian	25.0	54	37.0
Vietnamese	24.2	55	36.4
Chinese	23.8	47	42.6
Danish	21.7	54	44.4
Swedish	20.8	53	34.0
Basque	20.6	51	64.7
Indonesian	20.3	45	71.1
Latvian	17.0	52	42.3
Spanish	16.1	45	33.3
English	16.0	50	20.0
Bulgarian	15.6	52	46.2
Italian	14.1	48	56.2
Arabic	12.6	46	58.7
Hebrew	11.4	51	74.5
Persian	10.3	50	46.0
Hindi	8.4	51	41.2

Table 4: PDI statistics for UD 2.3 models, $b_{\text{avg}|\cdot|}$ metric, sorted by the mass metric (sum of PDIs). Agglutinative languages in **bold**, introflexive in *italics*.

Hyperparameters. The tagger is a hierarchical model, in which a character-level LSTM is used to create word level representations, which are fed into a second LSTM that predicts the word level tags. The character-level LSTM (in the baseline scenario) is bidirectional with a total hidden state size of 128 (64 units in each direction). For all models, the word-level LSTM is bidirectional with 2 layers and a hidden state size of 128, with 50% dropout applied in the style of Gal and Ghahramani (2016). The character embedding size is fixed throughout at 256. The hidden state of the word level LSTM is used to predict the tags using a separate MLP for each attribute. Each MLP has a single hidden layer that is the same size as the tagset size for that attribute, and includes a tanh nonlinearity. Models were trained for up to 80 epochs, and we select the model with the highest POS tagging accuracy on the dev set. Training used SGD with 0.9 momentum, and all models were implemented using DyNet 2.0 (Neubig et al., 2017).

Failed Models. To facilitate direct analysis of a weight matrix, we considered using a vanilla RNN as the character level encoding mechanism. Preliminary experiments indicated that this model would not reach acceptable tagging performance levels. Other preliminary experiments found that dropout at the word level seemed to greatly increase the stability of the model. English models could be trained for many epochs past convergence without signs of over-fitting, but only when using dropout.

D Additional Methods

Prior to our focus on hidden state analysis, we considered directly analyzing the weight matrix of an RNN (or the gate matrices in an LSTM). Our goal was to create a metric that would identify similar weight matrices, with the hope that models trained on languages with similar morphological properties would have similar matrices. One such candidate was the mathematical concept of matrix similarity (i.e. two matrices that represent the same linear operation after a possible change of basis for one). To test the suitability of this metric, we trained an English model using a vanilla RNN as the character encoder. We then applied a change of basis to the weight matrix (generating several mathematically similar matrices), and inserted these into new models as frozen parameters. The hope was that we could replace a weight matrix with a mathematically similar matrix and the model could learn to adapt to the similar matrix more quickly than if training from scratch. This would imply that a mathematically similar matrix is still useful to a network. Unfortunately, preliminary experiments did not suggest that this was the case, so we abandoned this line of inquiry.