Exploring Errors in POS-tagging by Quantifying Model Uncertainty

Michael Neely
University of Amsterdam
Student number: 12547190
✉ michael.neely

Stefan Schouten
University of Amsterdam
Student number: 12808113
✉ stefan.schouten

Leila Talha
University of Amsterdam
Student number: 10756922
✉ leila.talha

Abstract

Part-of-speech (POS) tagging is an import pre-processing step in Natural Language Processing. State-of-the-art neural approaches typically produce rich, context-sensitive word encodings with recurrent networks. A recently proposed and highly successful *meta* recurrent architecture integrates sentence-level context from both character and word-based representations. In this work, we exploit Bayesian model averaging to analyze the uncertainty of the different components of a recurrent meta-architecture in the context of POS tagging. We find that the meta component mediates the signals from the word and character-based components. Most importantly, we show that the meta model is highly uncertain when its input signals disagree.

1 Introduction

Part-of-speech (POS) tagging — labeling each word in a sentence with its grammatical role — is considered a nearly solved task in Natural Language Processing (NLP). Current efforts focus on obtaining the last few percentage points of word accuracy and the more challenging task of improving sentence accuracy (Manning, 2011). POS tagging is an essential pre-processing step for other NLP tasks, such as named-entity recognition (Ritter et al., 2011) and lemmatization (Straka and Straková, 2017). Therefore, it is worth being aware of contemporary POS taggers' strengths and weaknesses and how they might influence performance on the downstream task.

A distinction between two types of taggers can be made. More traditional methods for POS tagging tend to be rule-based, deploying manually constructed grammars to tag a given sentence correctly. However, with the surge of data-driven approaches in NLP, machine learning methods have become increasingly popular. The current state-of-the-art performance is reported at 97.85% token accuracy[1]. Therefore, we focus on the data-driven approach.

Specifically, we implement and train the meta-BiLSTM as proposed by Bohnet et al. (2018) because it competes with the current state-of-the-art through its novel combination of two bidirectional LSTMs: one sentence-based character model and another sentence-based word model. We additionally apply Monte Carlo (MC) drop-out to estimate the uncertainty over the predictions of the component and the meta model, which will form the basis of our analysis of the behaviour of the meta-BiLSTM.

We find that the uncertainty of the meta-BiLSTM decreases as the approximate predictive distributions of its character and the word-based components become more similar. Our results show that the meta-model typically assigns a probability distribution over tags that is similar to the sub-model with the least uncertainty. Therefore, we postulate that the meta model mediates input signals from its components to make a final prediction.

2 Related Work

Efforts to improve tagging performance by augmenting word representations with sub-word features are common. Tseng et al. (2005) exploit morphological features such as prefixes and suffixes to achieve state-of-the-art performance in Mandarin tagging with a maximum entropy Markov model.

Later, Dos Santos and Zadrozny (2014) introduce a model that learns character-level representations of words by association convolutions with traditional word-embeddings. Instead of using pre-trained word-embeddings, Ling et al. (2015) use a BiLSTM to learn character-level embeddings, which they feed to a compositional model that constructs vector representations of words - analogously to how humans build words from characters.

The work of Dozat et al. (2017) shows that a combination of both word and character-level representations (both LSTM-based) leads to an improvement in POS tagging accuracy. Their work is similar to that of Bohnet et al. (2018). The most notable difference is that the character-level representations they use are insensitive to a sentence's context because the recurrent units of the LSTM are restricted to word boundaries. In this respect, their work is similar to the other works laid out

[1] https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

above.

3 Methods

3.1 The meta-BiLSTM

The meta-BiLSTM architecture is composed of two component models and one meta model. The first component model, which we refer to as the *character model* produces sentence-level context sensitive word encodings from character embeddings. The second component model, which we refer to as the *word model* produces sentence-level context sensitive word encodings from word embeddings. Encodings from both component models are fed into the *meta model* which leverages their combined context sensitive encodings. We refer to the combination of all three models as the *meta-BiLSTM*.

Notably, the component and meta models are independent neural architectures capable of classifying POS tags in isolation of the other models. They can be optimized separately with their own loss functions or jointly optimized as a collective. The architecture of the meta-model is sketched in Figure 1 and the architectures of the character and word models are further detailed in paragraphs § 3.1.1 and § 3.1.2.

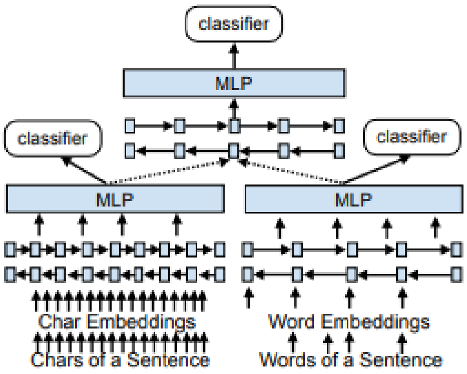

Figure 1: A schematic overview of the meta-BiLSTM from (Bohnet et al., 2018). The data flows along the arrows and losses are backpropagated over the bold lines.

3.1.1 Sentence-based character model

The character model takes as input sentences that are split into UTF8-characters. Spaces between the tokens, assuming the input has been tokenised, are included and each character is mapped to an initial, dynamically learned embedding.

Subsequently, a BiLSTM reads the character-embeddings from an n-character sentence $(e_i^{char}, \dots, e_n^{char})$ from left to right and vice versa:

$$f^0_{c,i}, b^0_{c,i} = \text{BiLSTM}(r_0, (e_i^{char}, \dots, e_n^{char}))_i$$

The character model contains l of these layers that feed into each other by concatenating the output encodings from previous layers, except for the final layer l, which has separate output vectors for each character $(f^l_{c,1}, \dots, f^l_{c,n})$ and $(b^l_{c,1}, \dots, b^l_{c,n})$ for both the forward and backward direction, respectively.

Finally, the character model concatenates the output vectors of the first and last character of a word w from both the forward and backward direction -i.e., concat$(F_{\text{1st}}(w), F_{\text{last}}(w), B_{\text{1st}}(w), B_{\text{last}}(w))$- to form word encodings. This process is illustrated in Figure 2. The resulting word encodings are passed to an MLP with on top a linear classifier, that outputs the most probable tag for each word.

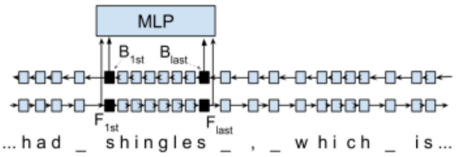

Figure 2: A schematic overview of the character model from (Bohnet et al., 2018). The four black boxes indicate the outputs for the example token `shingles`.

3.1.2 Sentence-based word model

The word model has an architecture that is similar to that of the character model but there are some differences: the input consists of word-encodings rather than character-encodings and for each word, pretrained embeddings $(p_1^{word}, \dots, p_n^{word})$ are summed with dynamically learned embeddings $(e_1^{word}, \dots, e_n^{word})$ -i.e., $in_i^{word} = p_i^{word} + e_i^{word}$.

Then, the summed embeddings are passed to l BiLSTM layers of which the output $f^l_{w,i}, b^l_{w,i}$ is again concatenated and used as input to an MLP with a linear classifier that predicts the part-of-speech tag with the highest assigned probability.

3.2 Meta model

For each word, the meta model concatenates the output of that word's context sensitive character and word-based encodings. This concatenation is passed through a BiLSTM and MLP classifier in the same fashion as the component models.

$$cw_i = \text{concat}(m_i^{char}, m_i^{word})$$

$$f^l_{m,i}, b^l_{m,i} = \text{BiLSTM}(r0, (cw_0, \dots, cw_n))_i$$

$$m_i^{comb} = \text{MLP}(\text{concat}(f_{m,i}^l, b_{m,i}^l))$$

The predicted tag of the meta model is considered as the prediction of the entire meta-BiLSTM. The predictions of the component models are also available if necessary.

3.3 Monte Carlo drop-out

Generally, drop-out is a training technique that switches off neurons in a neural network according to some probability parameter p. This essentially results in a different network at each training step whilst the objective function remains unchanged and thus reduces the chances of overfitting. When this technique is applied during both training and testing, it is called *Monte Carlo drop-out*.

When doing several forward passes with the same test data, the distribution of predictions made by each of the slightly altered models due to drop-out can be used to estimate the mean and variance of the predictive distribution. The variation of the predictive distribution is an estimate of the model's uncertainty. Thus, Monte Carlo drop-out approximates Bayesian variational inference (Gal and Ghahramani, 2016).

4 Analysis

4.1 Experimental set-up

The meta-BiLSTM has been implemented using the AllenNLP platform (Gardner et al., 2017). Both the character and word model have three BiLSTM layers with hidden states of size 400, whereas the meta model merely has one. The drop-out probability for the LSTM, MLP and embedding units is 0.333 -except for the embedding unit of the character model for which $p = 0.05$. The non-linear activation of the neurons in each of the three MLPs is the ELU function.

The word-embeddings are initialised with zero values and the pre-trained embeddings remain unchanged. On the other hand, the character-embeddings and the model parameters of the MLPs are initialised according to a standard Gaussian.

We train the meta-BiLSTM on the CoNLL-2000 dataset for a maximum of 40 epochs with a patience value of 5 epochs for early stopping. We hold out ten percent of the training for validation. We follow the advice of Bohnet et al. (2018) and optimize the meta, character, and word models separately. Each sub-model is optimized to minimize the cross-entropy loss with the AMSGrad variant (Tran and Phong, 2019) of Adam (Kingma and Ba, 2017). We choose the same hyperparameters as Bohnet et al. (2018), namely: a learning rate of 0.02, weight decay of 0.999994, an epsilon of $1e-8$ and respective beta values of 0.9 and 0.999.

4.2 Results

Model Component	Accuracy
Character	0.966355
Word	0.983135
Meta	0.985626

Table 1: Average meta-BiLSTM model component accuracy on the CoNLL2000 test set

Uncertainty per POS Tag - Table 1 details the accuracy of each component of the meta-BiLSTM model. As expected, the accuracy of the meta component is higher than the word and character-based sub-components.

Table 2 shows the uncertainty. We can see that generally the meta-model is less uncertain about its predictions than the character and word models. This is an indication that the meta-model is successfully combining information from both of the models.

Overall we see that the rarest tags in the training set are among the highest in terms of uncertainty. This is expected and even desirable from a variational model. There are a number of exceptions though. Mostly these are tags that are very specific, for example all the tags for specific kinds of punctuation (not displayed in Table 2 but can be seen in Appendix B). Other very specific tags are the tag for 'Existential there' (EX) and for the word 'to' (TO). Slightly less specific but still easily identifiable are those words that fall under the 'Interjection' (UH) tag. These are things people say to express reactions or embellish an otherwise bland utterance, for example: 'ouch!', or 'huh?'. We see that for this tag the word model is more confident than the character model. This is likely because the word model has the ability to learn word-specific representations. The superlative form of adverbs are easily identifiable for similar reasons. Although for this tag the character model is equally confident, perhaps because it can recover the tag from the particular suffixes that superlatives often display. Finally the Wh-pronouns (WP$) are predicted with high confidence by the character model, most likely because of their defining characteristic: the fact that they all start with 'wh'.

Tag	%	Model		
		Char	Word	Meta
CC	2.537%	0.00279	0.00089	0.00127
CD	3.927%	0.00579	0.00792	0.01769
DT	8.660%	0.00497	0.00264	0.00244
EX	0.097%	0.01064	0.00842	0.01734
FW	0.018%	0.13529	0.19578	0.20562
IN	10.752%	0.01232	0.00457	0.00506
JJ	6.180%	0.08770	0.03958	0.04628
JJR	0.403%	0.10107	0.05170	0.04268
JJS	0.177%	0.03304	0.02795	0.03573
MD	1.023%	0.00368	0.00647	0.00559
NN	14.239%	0.05989	0.02348	0.02857
NNP	9.391%	0.03376	0.02593	0.03580
NNPS	0.198%	0.15715	0.11632	0.10179
NNS	6.432%	0.03490	0.01821	0.03269
PDT	0.026%	0.19268	0.17268	0.17655
POS	0.836%	0.00551	0.00251	0.00516
PRP	1.804%	0.00547	0.00239	0.00466
PRP$	0.888%	0.00214	0.00359	0.00335
RB	3.121%	0.05379	0.02647	0.03256
RBR	0.152%	0.12757	0.06562	0.07460
RBS	0.090%	0.00415	0.00593	0.00604
RP	0.039%	0.25361	0.18869	0.13567
TO	2.400%	0.00000	0.00003	0.00050
UH	0.007%	0.12484	0.00137	0.00116
VB	2.842%	0.04186	0.02787	0.02925
VBD	3.186%	0.04860	0.02014	0.02261
VBG	1.545%	0.07095	0.03746	0.04651
VBN	2.250%	0.07681	0.05035	0.05244
VBP	1.355%	0.06326	0.03325	0.03143
VBZ	2.195%	0.04242	0.02032	0.02332
WDT	0.451%	0.03640	0.02574	0.03188
WP	0.250%	0.00924	0.00291	0.00662
WP$	0.017%	0.00035	0.09507	0.09307
WRB	0.226%	0.01158	0.01195	0.01127

Table 2: Mean uncertainty of the character, word and meta models per part-of-speech tag.

We also see that generally, the character model is more uncertain than the word model. Interestingly this is not true for the foreign word (FW) tag. An hypothesis for this observation is that the foreign words come from languages that are morphologically richer than English. To be able to correctly predict such words, the word model must have seen the exact same surface form in the train data, whereas the character model may rely on previously seen affixes.

Char	Word	None
275	1422	76

Table 3: When there is disagreement among the models, the meta-model most often sides with the word-model.

Table 3 shows that when the two component models disagree the meta model seldom predicts an alternative POS tag. This indicates that the meta-BiLSTM functions as some sort of mediator between the two component models.

Our results indicate that the meta-BiLSTM balances between the predictions of its character-based and word-based components. This is supported by the observation that the uncertainty over the predictions of the meta-BiLSTM decreases with the Jensen Shannon divergence between the approximate predictive distributions of the character and word models - i.e., their distributions become more similar- as is illustrated in Figure 3. We measure a Pearson correlation of 0.687, with $p < 0.01$.

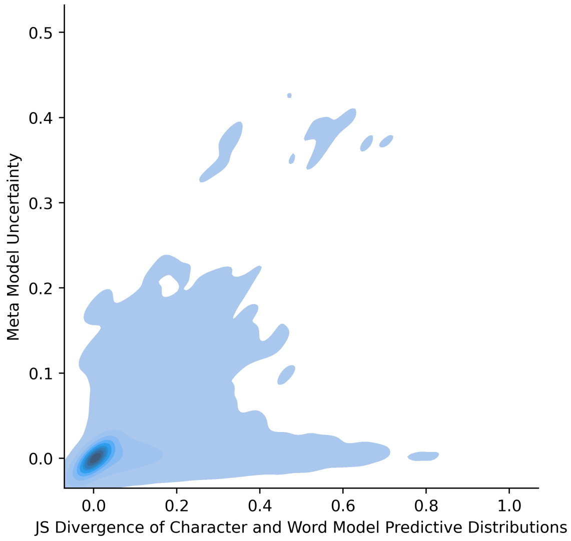

Figure 3: Uncertainty in the meta model as a function of the disagreement of the character and word-based models.

We further observe an interesting pattern, for cases where there is disagreement between the models the meta-model seems to assign probabilities that are closest to those assigned by the sub-model with least uncertainty, this pattern is highlighted further in Appendix C.

5 Conclusion

The objective of this project was to analyse the behaviour of the meta-BiLSTM with respect to the signals received from its component models. Our main finding is that the uncertainty of the meta-BiLSTM increases with the Jensen-Shannon divergence between the approximate predictive distributions of the component models. We hope that our analysis of the model's uncertainty inspires future architectural improvements.

References

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Cicero Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *International Conference on Machine Learning*, pages 1818–1826. PMLR.

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.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

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. *arXiv preprint arXiv:1508.02096*.

Christopher D. Manning. 2011. Part-of-speech tagging from 97linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I*, CICLing'11, page 171–189, Berlin, Heidelberg. Springer-Verlag.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1524–1534.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99.

Phuong Thi Tran and Le Trieu Phong. 2019. On the convergence proof of amsgrad and a new version. *IEEE Access*, 7:61706–61716.

Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. 2005. Morphological features help pos tagging of unknown words across language varieties. In *Proceedings of the fourth SIGHAN workshop on Chinese language processing*.

A Uncertainty per POS tag - Visualisation

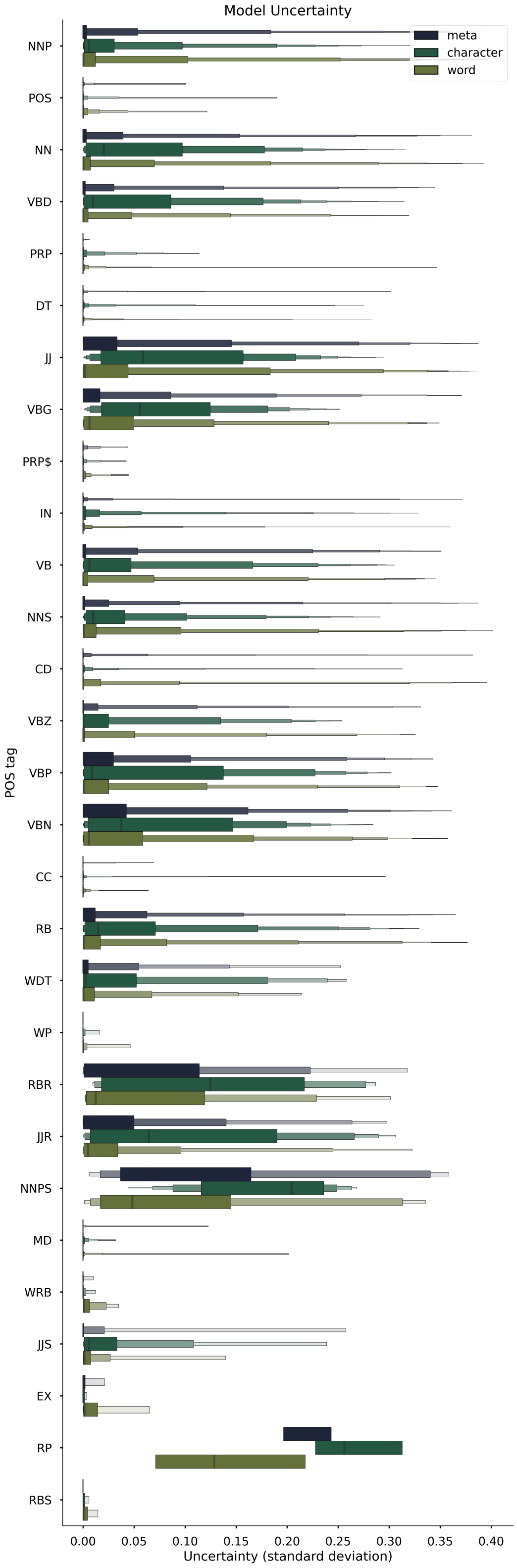

Figure 4: Uncertainty for a selection of POS tags for each model as it varies for different instances.

B Estimates and uncertainty for all tags.

tag	%	Estimate			Uncertainty		
		character	meta	word	character	meta	word
,	5.087%	1.00000	1.00000	0.99998	0.00004	0.00000	0.00027
:	0.495%	0.99995	1.00000	0.99941	0.00054	0.00001	0.00470
.	4.169%	0.99993	1.00000	0.99988	0.00037	0.00000	0.00070
”	0.705%	0.99375	0.99415	0.99464	0.00195	0.00098	0.00273
(	0.129%	0.99999	0.99998	0.99956	0.00008	0.00033	0.00349
)	0.133%	0.99958	0.99855	0.98625	0.00379	0.01001	0.04700
#	0.017%	0.99965	0.99767	0.99642	0.00396	0.02438	0.02163
“	0.723%	0.99999	1.00000	0.99987	0.00022	0.00003	0.00092
$	0.827%	0.99992	0.99997	0.99969	0.00044	0.00049	0.00125
CC	2.537%	0.99826	0.99982	0.99981	0.00279	0.00089	0.00127
CD	3.927%	0.99309	0.99429	0.97775	0.00579	0.00792	0.01769
DT	8.660%	0.99376	0.99770	0.99780	0.00497	0.00264	0.00244
EX	0.097%	0.99304	0.99848	0.99559	0.01064	0.00842	0.01734
FW	0.018%	0.40541	0.29865	0.50616	0.13529	0.19578	0.20562
IN	10.752%	0.98664	0.99538	0.99484	0.01232	0.00457	0.00506
JJ	6.180%	0.82529	0.95737	0.94246	0.08770	0.03958	0.04628
JJR	0.403%	0.84731	0.96335	0.96909	0.10107	0.05170	0.04268
JJS	0.177%	0.96991	0.97972	0.96262	0.03304	0.02795	0.03573
MD	1.023%	0.98486	0.99322	0.99476	0.00368	0.00647	0.00559
NN	14.239%	0.89247	0.96592	0.96149	0.05989	0.02348	0.02857
NNP	9.391%	0.94980	0.97151	0.95871	0.03376	0.02593	0.03580
NNPS	0.198%	0.60176	0.74729	0.71977	0.15715	0.11632	0.10179
NNS	6.432%	0.95556	0.97964	0.96699	0.03490	0.01821	0.03269
PDT	0.026%	0.81733	0.79115	0.77497	0.19268	0.17268	0.17655
POS	0.836%	0.99778	0.99933	0.99623	0.00551	0.00251	0.00516
PRP	1.804%	0.98904	0.99314	0.99636	0.00547	0.00239	0.00466
PRP$	0.888%	0.99940	0.99816	0.99828	0.00214	0.00359	0.00335
RB	3.121%	0.93304	0.97728	0.97023	0.05379	0.02647	0.03256
RBR	0.152%	0.78103	0.93656	0.90733	0.12757	0.06562	0.07460
RBS	0.090%	0.97912	0.97895	0.97905	0.00415	0.00593	0.00604
RP	0.039%	0.68689	0.52263	0.55567	0.25361	0.18869	0.13567
TO	2.400%	1.00000	1.00000	0.99995	0.00000	0.00003	0.00050
UH	0.007%	0.42648	0.49978	0.50011	0.12484	0.00137	0.00116
VB	2.842%	0.90907	0.95598	0.95560	0.04186	0.02787	0.02925
VBD	3.186%	0.92355	0.97296	0.97099	0.04860	0.02014	0.02261
VBG	1.545%	0.89485	0.95363	0.93894	0.07095	0.03746	0.04651
VBN	2.250%	0.83876	0.93020	0.92922	0.07681	0.05035	0.05244
VBP	1.355%	0.86263	0.95169	0.95235	0.06326	0.03325	0.03143
VBZ	2.195%	0.89879	0.97184	0.96813	0.04242	0.02032	0.02332
WDT	0.451%	0.94371	0.97451	0.96718	0.03640	0.02574	0.03188
WP	0.250%	0.99611	0.99958	0.99880	0.00924	0.00291	0.00662
WP$	0.017%	0.99994	0.98421	0.95071	0.00035	0.09507	0.09307
WRB	0.226%	0.99080	0.98632	0.98771	0.01158	0.01195	0.01127

Table 4: On the left we have the mean probability assigned by each model to the labels for each word. On the right we have the corresponding standard deviations (uncertainty) of each model.

C Analysis of disagreements.

	tag	#	Estimate			Uncertainty		
			character	meta	word	character	meta	word
*	CD	37	0.8031	0.7850	0.3193	0.0568	0.2452	0.2178
*	DT	15	0.3033	0.7423	0.7668	0.1762	0.2109	0.1585
*	FW	1	0.0039	0.1466	0.5417	0.0090	0.1938	0.3716
*	IN	37	0.3957	0.7311	0.7364	0.2394	0.1685	0.1162
*	JJ	383	0.3254	0.8123	0.7737	0.1504	0.1388	0.1204
*	JJR	22	0.2711	0.7542	0.8178	0.1976	0.2402	0.1841
*	JJS	4	0.6836	0.6442	0.4206	0.2034	0.3962	0.3656
	MD	7	0.0350	0.5509	0.6627	0.1043	0.3966	0.2904
*	NN	408	0.3415	0.7414	0.7298	0.1704	0.1585	0.1378
*	NNP	188	0.5274	0.5953	0.4971	0.1511	0.2158	0.1744
*	NNPS	43	0.3891	0.6161	0.5398	0.1741	0.1722	0.1477
	NNS	83	0.5304	0.5595	0.4761	0.1633	0.1985	0.1449
*	POS	1	1.0000	0.8274	0.0658	0.0000	0.3326	0.1881
*	PRP	8	0.0164	0.3034	0.6513	0.0284	0.2269	0.3171
*	RB	56	0.3358	0.7622	0.7607	0.1737	0.1860	0.1546
*	RBR	14	0.3722	0.7970	0.7513	0.2238	0.1947	0.1380
	RBS	1	0.0010	0.0000	0.0006	0.0128	0.0001	0.0033
*	RP	4	0.6789	0.2666	0.3178	0.2855	0.1887	0.1313
	VB	81	0.2576	0.5959	0.6175	0.1467	0.2266	0.2022
*	VBD	78	0.4148	0.6975	0.7130	0.1755	0.1834	0.1536
*	VBG	47	0.5839	0.5001	0.4091	0.1687	0.2350	0.1708
*	VBN	123	0.3411	0.6990	0.7293	0.1683	0.1925	0.1534
*	VBP	47	0.2148	0.7039	0.7392	0.1504	0.1683	0.1281
*	VBZ	78	0.2961	0.8324	0.8305	0.1714	0.1317	0.1210
	WDT	6	0.4209	0.6321	0.5750	0.2004	0.2304	0.2223
*	WRB	1	0.4864	0.1588	0.0450	0.3861	0.3129	0.1333

Table 5: Same estimate and uncertainty but now only for those instances where there was disagreement among the three models about which tag to predict. Note that the meta model usually agrees with the word model, sometimes even when this is not beneficial. We observe an interesting effect: the meta model seems to assign similar probabilities as the sub-model that is least uncertain, ”it sides with the most confident sub-model”. This holds for each row that is marked with a star.