The_Illiterati: Part-of-Speech Tagging for Magahi and Bhojpuri without Even Knowing the Alphabet

1st Thomas Proisl
Institute of Cognitive Science
Osnabrück University
Osnabrück, Germany
thomas.proisl@uos.de

2nd Peter Uhrig
English and American Studies
FAU Erlangen-Nürnberg
Erlangen, Germany
peter.uhrig@fau.de

3rd Philipp Heinrich
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
Erlangen, Germany
philipp.heinrich@fau.de

4th Andreas Blombach
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
Erlangen, Germany
andreas.blombach@fau.de

5th Sefora Mammarella
Romance Studies
FAU Erlangen-Nürnberg
Erlangen, Germany
sefora.mammarella@icloud.com

6th Natalie Dykes
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
Erlangen, Germany
natalie.mary.dykes@fau.de

7th Besim Kabashi
Computational Corpus Linguistics
FAU Erlangen-Nürnberg
Erlangen, Germany
besim.kabashi@fau.de

***Abstract*—In this paper, we describe the part-of-speech-tagging experiments for Magahi and Bhojpuri that we conducted for our participation in the NSURL 2019 shared tasks 9 and 10 (Low-level NLP Tools for (Magahi|Bhojpuri) Language). We experiment with three different part-of-speech taggers and evaluate the impact of additional resources such as Brown clusters, word embeddings and transfer learning from additional tagged corpora in related languages. In a 10-fold cross-validation on the training data, our best-performing models achieve accuracies of 90.70% for Magahi and 94.08% for Bhojpuri. Accuracy increased to 94.79% for Magahi and dropped to 78.68% for Bhojpuri on the test data.**

***Index Terms*—natural language processing, part-of-speech tagging, under-resourced languages**

I. Introduction and Related Work

Magahi and Bhojpuri are two of the three principal languages of the Bihari group (Maithili being the third). There are competing categorizations of the Bihari group within the Indo-Aryan languages (see [3]–[5]). While there are few Magahi speakers outside of Southern Bihar, Bhojpuri is spoken in parts of two Indian states, Western Bihar and Eastern Uttar Pradesh, and the Southwest of Nepal. According to the 2011 census [7], about 51 million people in India stated Bhojpuri as their mother tongue, and about 13 million did so for Magahi. However, these numbers may seriously underestimate the actual number of speakers, since speakers of both languages often name Hindi as their first language – the language used in schools, courts, and other public institutions [15, p. 547].

Despite these numbers, comparatively few linguistic resources and NLP tools currently exist for both languages, with most of the scarce attention having gone towards Bhojpuri (e.g. [8]).

It is beyond the scope of this paper and our own expertise to describe both languages in detail (but see, e.g., [14], [15]). Among the features which appear pertinent to part-of-speech tagging of Magahi and Bhojpuri are SOV order, rich verb morphology, the extensive use of postpositions, and the unusual agreement system of Magahi (where the verb has to agree with subject and object simultaneously).

Table I gives an overview of the two datasets of the shared task. While the training set for Bhojpuri is much larger, it also features a more fine-grained tagset.

	Magahi	Bhojpuri
training	61.435	94.692
test	8.205	10.582
tagset size	18	33

TABLE I
Sizes of the training and test sets and of the tagsets.

II. Strategies and Systems

A. Part-of-Speech Taggers

We experiment with three different, freely available part-of-speech taggers:

- SoMeWeTa [10], a tagger based on the averaged structured perceptron that supports domain adaptation and

can incorporate external sources of information such as Brown clusters.[1]
- A BiLSTM+CRF sequence tagger by Guillaume Genthial that uses character and word embeddings and supports transfer learning.[2]
- The Stanford Tagger [12], which is based on a maximum entropy cyclic dependency network.[3]

B. Additional Resources

In addition to the training data provided by the task organizers, we use the following freely available resources:

- The Hindi UD treebank, which is based on the Hindi Dependency Treebank (HDTB; ca. 352,000 tokens) [1], [9].[4]
- A POS-tagged Magahi corpus (KMI-Mag; ca. 46,000 tokens) and a corpus of untagged Magahi texts (ca. 2.8 million tokens).[5]
- Wikimedia dumps for Hindi (ca. 34.7 million tokens) and Bihari (ca. 700,000 tokens).[6] We extract the plain text using wikiextractor[7] and tokenize and sentence-split it using the ICU tokenizer via polyglot.[8]
- Brown clusters [2] computed from the tokenized Wikimedia dumps and the untagged Magahi corpus (1000 clusters, minimum frequency 5).[9]
- Pre-trained fastText embeddings for Hindi and Bihari[10]

The additional tagged Magahi corpus (KMI-Mag) is annotated with a tagset consisting of 35 tags which is almost identical to the 33-tag tagset used in the Bhojpuri corpus. KMI-Mag uses three tags that do not occur in the Bhojpuri data (`V_VM_VF`, `V_VM_VNF` and `V_VM_VNP`) and misses one tag that is used for Bhojpuri (`RD_ECH_B`). For our transfer learning experiments targeting Bhojpuri, we simply convert the three verb tags to `V_VM`. For targeting Magahi, we map the 35 tags to UD tags.

C. Experiments using SoMeWeTa

The distinctive features of SoMeWeTa are its ability to leverage additional resources and its transfer learning or domain adaptation capabilities. Consequently, we focus on these two aspects in our experiments.

For Bhojpuri, we experiment primarily with the Brown clusters computed from the Hindi and Bihari Wikimedia dumps and the untagged additional Magahi corpus (cf. section II-B). Our cross-validation experiments show that the Brown clusters have a small positive effect with the best results being obtained by Brown clusters computed from the union of all three additional corpora (cf. Table II). With KMI-Mag we have a corpus of a closely related language that is annotated with an almost identical tagset (cf. section II-B). However, pretraining on that and then adapting to Bhojpuri seems to have no noticeable effect.

model	accuracy
No additional resources	91.62 (±0.97)
Hindi Brown clusters	91.79 (±1.00)
Bihari Brown clusters	91.60 (±1.01)
Magahi Brown clusters	91.69 (±0.93)
Hindi+Magahi Brown clusters (hi+mag)	91.99 (±0.83)
Hindi+Bihari+Magahi Brown clusters (hi+bh+mag)	*92.04 (±0.80)*
KMI-Mag → Bhojpuri, hi+mag	92.03 (±0.90)
KMI-Mag → Bhojpuri, hi+bh+mag	92.06 (±0.94)

TABLE II
BHOJPURI RESULTS FOR SOMEWETA. WE REPORT THE MEAN ACCURACIES AND 95% CONFIDENCE INTERVALS OF A 10-FOLD CROSS-VALIDATION ON THE TRAINING DATA. THE MODEL THAT WE SUBMITTED TO THE SHARED TASK IS SET IN ITALICS.

For Magahi, we experiment with a wide range of transfer learning settings in addition to the different Brown clusters:

- Pretraining on one of KMI-Mag, HDTB or the Bhojpuri dataset (mapped to UD tags).
- Pretraining on all possible combinations of KMI-Mag, HDTB and the Bhojpuri dataset (using the concatenation of these resources).
- Longer pretraining chains where we start with HDTB and adapt to one or two other resources before we do the final adaptation to Magahi.

The best results are obtained by using Brown clusters computed from the Hindi Wikimedia dumps and the untagged additional Magahi corpus. As for Bhojpuri, transfer learning does not seem to have any noticeable effect (cf. Table III).

D. Experiments using the BiLSTM-CRF tagger

Neural networks with a BiLSTM-CRF architecture achieve POS-tagging results close to the current state of the art.[11] In our experiments, we focus less on the hyperparameters of the network but on the effects of our additional resources. We try out both the Hindi and Bihari fastText embeddings. Since the Bihari embeddings do not perform significantly better than the Hindi embeddings (cf. Table IV) and the Hindi embeddings cover a much larger vocabulary (15.3 million instead of 8.9 million), we use the Hindi embeddings for our further experiments. For these experiments, we make use of the transfer learning abilities of the tagger and pretrain the models on HDTB or KMI-Mag. The BiLSTM-CRF tagger seems to benefit more from the transfer learning setting than SoMeWeTa and achieves its best results for both languages with a transfer from KMI-Mag. Interestingly, the BiLSTM-CRF can outperform SoMeWeTa only on the Magahi dataset while it performs notably worse on the Bhojpuri dataset.

[1] https://github.com/tsproisl/SoMeWeTa
[2] We use the slightly modified version by [11]: https://github.com/riedlma/sequence_tagging
[3] https://nlp.stanford.edu/software/tagger.html
[4] https://github.com/UniversalDependencies/UD_Hindi-HDTB/tree/master
[5] https://github.com/kmi-linguistics/magahi
[6] https://dumps.wikimedia.org
[7] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor
[8] http://polyglot-nlp.com/
[9] We use the implementation by [6]: https://github.com/percyliang/brown-cluster
[10] https://fasttext.cc/docs/en/crawl-vectors.html
[11] Cf. https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

model	accuracy
No additional resources	88.92 (±1.24)
Hindi Brown cluster	89.07 (±1.24)
Bihari Brown cluster	88.90 (±1.32)
Magahi Brown cluster	89.12 (±1.23)
Hindi+Magahi Brown cluster	*89.32 (±1.15)*
Hindi+Bihari+Magahi Brown cluster	89.15 (±1.17)
KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.20 (±1.10)
KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.23 (±1.19)
Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.25 (±1.13)
Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.18 (±1.25)
HDTB → Magahi, Hindi+Magahi Brown cluster	89.26 (±1.21)
HDTB → Magahi, Hindi+Bihari+Magahi Brown cluster	89.17 (±1.18)
HDTB+KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.22 (±1.12)
HDTB+KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.19 (±1.23)
HDTB+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.23 (±1.13)
HDTB+Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.18 (±1.20)
KMI-Mag+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.30 (±1.14)
KMI-Mag+Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.06 (±1.19)
HDTB+KMI-Mag+Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.21 (±1.17)
HDTB+KMI-Mag+Bhojpuri, Hindi+Bihari+Magahi Brown cluster	89.20 (±1.20)
HDTB → KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.24 (±1.20)
HDTB → KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.22 (±1.18)
HDTB → Bhojpuri → Magahi, Hindi+Magahi Brown cluster	89.27 (±1.14)
HDTB → Bhojpuri → Magahi, Hindi+Bihari+Magahi Brown cluster	89.11 (±1.17)
HDTB → Bhojpuri → KMI-Mag → Magahi, Hindi+Magahi Brown cluster	89.22 (±1.11)
HDTB → Bhojpuri → KMI-Mag → Magahi, Hindi+Bihari+Magahi Brown cluster	89.20 (±1.19)

TABLE III
MAGAHI RESULTS FOR SOMEWETA. WE REPORT THE MEAN ACCURACIES AND 95% CONFIDENCE INTERVALS OF A 10-FOLD CROSS-VALIDATION ON THE TRAINING DATA. THE MODEL THAT WE SUBMITTED TO THE SHARED TASK IS SET IN ITALICS.

model	accuracy
Magahi (Hindi embeddings)	88,97 (±1,14)
Magahi (Bihari embeddings)	89,09 (±1,00)
HDTB → Magahi (Hindi embeddings)	89,85 (±0,99)
KMI-Mag → Magahi (Hindi embeddings)	*90,70 (±0,92)*
Bhojpuri (Hindi embeddings)	90,78 (±0,55)
Bhojpuri (Bihari embeddings)	90,80 (±0,57)
KMI-Mag → Bhojpuri (Hindi embeddings)	*91,23 (±0,68)*

TABLE IV
RESULTS FOR THE BILSTM-CRF TAGGER. WE REPORT THE MEAN ACCURACIES AND 95% CONFIDENCE INTERVALS OF A 10-FOLD CROSS-VALIDATION ON THE TRAINING DATA. THE MODELS THAT WE SUBMITTED TO THE SHARED TASK ARE SET IN ITALICS.

E. Experiments using the Stanford Tagger

The Stanford Log-linear Part-Of-Speech Tagger ([12], [13]) is a mature and stable tagger that still exhibits competitive performance. The system is feature-rich and offers a range of configuration options, the effects of which were not fully understood by our research group at the beginning. It was thus decided to run extensive brute-force hyperparameter tuning using educated guesses about the value ranges of the various parameters. The documentation in the JavaDoc for the MaxentTagger class[12] provides the necessary information. It was decided to set the following parameters with the values or ranges given in Table V and Table VI.

[12]https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/MaxentTagger.html

Combining all the parameters results in a total of 76,800 parameter combinations per language. Even though training and testing can be done in approximately 2 minutes on a recent personal computer, the sheer number of parameter combinations necessitated running the experiments on High-Performance-Computing infrastructure. The setup consisted of a central queue of filenames of property files that all clients involved subscribed to.

For Magahi, only two runs with all parameter combinations were performed, one with the the top 80% of the training data as actual training data and the bottom 2% as test data and one with the bottom 80% as training data and the top 20% as test data. The values discussed below are the arithmetic mean of the accuracies of those two runs. With the Magahi tagset being Universal-Dependencies-compliant, it was straightforward for us to identify closed class words by pos tag and supply the list to the tagger during the training phase.

For Bhojpuri, a full 10-fold cross-validation was carried out for each of the parameter combinations, so the averages discussed below are most likely more reliable than the ones for Magahi. Since the Bhojpuri tagset was more complicated, we decided in favour of having the computer learn the closed class tags automatically based on the default *closedClassTagThreshold* of 40, i.e. if more than 40 different words obtain the same pos tag, it is not considered a closed class.

Given that the training dataset is smaller than what is available for more commonly researched languages, we expected that for most thresholds, values below the default values might be more relevant than above, and that is why our choice of

parameter	default value	value/range
closedClassTags	(none)	ADP AUX CCONJ DET NUM PART PRON SCONJ PUNCT
arch - architecture	generic	generic, left3word, bidirectional5words
arch - further unknown-words option	(none)	naacl2003unknowns
arch - unicode shapes for rare words	(none)	unicodeshapes(-2,2), unicodeshapes(-1,1), unicodeshapes(0), (none)
iterations	100	100
learnClosedClassTags	false	false
curWordMinFeatureThresh	2	1..4
minFeatureThresh	5	1..5
rareWordMinFeatureThresh	10	1..10
rareWordThresh	5	1..8
veryCommonWordThresh	250	100, 150, 200, 250

TABLE V
SETTINGS AND PARAMETERS WITH RANGES FOR THE TRAINING OF THE STANFORD POS TAGGER FOR MAGAHI.

parameter	default value	value/range
closedClassTags	(none)	(none)
arch - architecture	generic	generic, left3word, bidirectional5words
arch - further unknown-words option	(none)	naacl2003unknowns
arch - unicode shapes for rare words	(none)	unicodeshapes(-2,2), unicodeshapes(-1,1), unicodeshapes(0), (none)
iterations	100	100
learnClosedClassTags	false	true
closedClassTagThreshold	40	40
curWordMinFeatureThresh	2	1..4
minFeatureThresh	5	1..5
rareWordMinFeatureThresh	10	1..10
rareWordThresh	5	1..8
veryCommonWordThresh	250	100, 150, 200, 250

TABLE VI
SETTINGS AND PARAMETERS WITH RANGES FOR THE TRAINING OF THE STANFORD POS TAGGER FOR BHOJPURI.

parameter values is skewed towards smaller numbers.

For both languages, performance decreases abruptly when *rareWordThresh* is set to 1. We exclude this setting for the remainder of the analysis, since it is obviously beneficial for the tagger to treat hapax legomena as rare words. Additionally, performance was insensitive to variation in *veryCommonWordThresh* since this value is actually ignored by the Tagger in our case. We thus fix this value at 250 and use simple linear models without interaction for analyzing the influence of all other variables on performance measures:

$$\text{acc.} = \beta_0 + \beta_1(\textit{unicodeshape}) + \beta_2(\textit{macro}) + \sum_{j=3}^{6} \beta_j \gamma_j + \varepsilon$$

where β_i are the coefficients, γ_j is one of the integer features (*rareWordThresh*, *curWordMinFeatureThresh*, *minFeatureThresh*, *rareWordMinFeatureThresh*), and ε is the residual error.

Accuracy for Bhojpuri varies around $\mu \approx 93.88$ with a standard deviation of approximately 0.064 and the linear model yields an adjusted R^2 of approximately 0.80. For Magahi, performance is overall lower ($\mu \approx 87.66$) and variation higher ($\sigma \approx 0.51$), yet this variation is well-explained by the linear model (adjusted $R^2 \approx 0.98$).

For both languages, the *macro* parameter has most influence on accuracy. For Bhojpuri, the best *macro* to use is `bidirectional5words` (yielding ceteris paribus 0.09 and 0.12 better results compared to `generic` and `left3words`, respectively). For Magahi, however, `generic` and `left3words` yield better results (both approximately 1.0 accuracy points better than `bidirectional5words`). This is rather surprising, since according to the authors of the Stanford Tagger, "[t]he `left3words` architectures are faster, but slightly less accurate, than the `bidirectional` architectures."[13] The only viable explanation that comes to mind for such a result is that possibly the Magahi gold standard corpus was annotated with a trigram tagger and not sufficiently corrected manually. This is in line with our observation that in the Magahi data, there were dubious tags attached to what should have been classified as punctuation marks, e.g. the grave accent (`) which was tagged five times as a noun, twice as an adposition, once as a verb and once as an auxiliary but only twice as punctuation.

Looking at the respective best-performing *macro* only, *rareWordThresh* explains most of the remaining variation, with a significant regression coefficient of about 0.02 for Bhojpuri and 0.07 for Magahi. The effect might however decrease for higher values than the ones tested here (*rareWordThresh* $\in \{1, \ldots, 8\}$).

unicodeshape has a small effect on performance for Bhojpuri, where `(-1,1)` and `(-2,2)` yield an increase in

[13]https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/ExtractorFrames.html

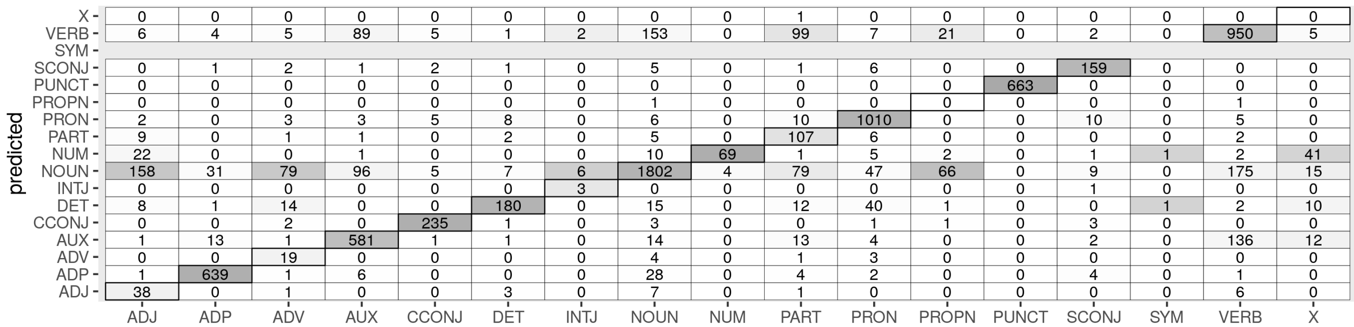

Fig. 1. Confusion Matrix for SoMeWeTa predicting Magahi tags on the test data. Absolute numbers are given for all cells; shade represents recall (on the diagonal) and false positive rate rate, respectively. Actual labels can be found on the abscissa, predicted ones on the ordinate.

performance of about 0.06 compared to `(0)` and `None`. This effect cannot be confirmed for Magahi. For both languages, performance decreases in *curWordThresh*, *curWordMinFeatureThresh*, and *rareWordMinFeatureThresh*, though the effect is negligible and not always significant. In both cases, *minFeatureThresh* does not have a significant influence on accuracy.

III. Results and Error Analysis

A. Bhojpuri

The overall results for Bhojpuri are delightful since they are even better than on our training data (see Table VIII): Our optimized version of the Stanford tagger scored 95 points macro F_1 (94.78 accuracy), and we thus share first place with our sole competitor (team *NITK-NLP*); SoMeWeta and the BiLSTM tagger are close behind.

rank	submission	F_1
1	**Stanford**	95
1	NITK-NLP_SUB1	95
2	**SoMeWeTa**	93
3	**BiLSTM-CRF**	92
4	NITK-NLP_SUB2	89

TABLE VII
Results for Bhojpuri

We omit the very large confusion matrix (33×33 and predominantly zero off the diagonal) and just provide a quick summary for the Stanford tagger here:[14]

- Two tags are not predicted at all by our tagger: `RD_ECH_B` (which appears once in the gold data and was misclassified as `N_NN` by our tagger), and `RD_UNK` (twice, once classified as `N_NN`, once as `V_VM`).
- `RP_INJ` appeared five times in the gold standard and was predicted four times correctly. This tag yields worst recall (apart from the two pathological cases above).
- 30 of the 195 occurrences of `RD_SYM` were misclassified (recall 84.6%), most prominently as `N_NN` (26 cases).
- Further incorrect predictions as `N_NN` occur for `JJ` (11.3% of its occurrences classified as `N_NN`, 85.2% recall), `RB` (7.7%, 89.7% recall), and `N_NNP` (6.4%, 92.8% recall).
- Another notable confusion is the pair `V_VM` (87.8% recall) and `V_VAUX` (86.6% recall); `V_VM` was predicted 64 times as `V_VAUX`, `V_VAUX` 66 times as `V_VM`. Also `V_VM` was predicted as N_NN 85 times.

The results for our other submissions were very much in line with the results discussed here.[15] All in all, the errors made by our submissions are very much what one would expect: Very rare categories are sometimes misclassified, very frequent categories (such as `N_NN`) tend to be the go-to label for misclassifications, and similar morphosyntactic categories are confused with one another (`V_VM` and `V_AUX`, `N_NN` and `N_NNP`).

B. Magahi

With a macro F_1 score of only 77%, our best submissions, SoMeWeTa (78.68 accuracy) and BiLSTM-CRF (78.86 accuracy), rank second in the task of predicting Magahi tags, closely behind one of our competing team's submissions. Results are peculiar, since this is a drop of more than ten points compared to our cross-validation on the training data set and way outside our realized confidence intervals (see Table III).

rank	submission	F_1
1	NITK-NLP_SUB2	79
2	**SoMeWeTa**	77
2	**BiLSTM-CRF**	77
3	**Stanford**	74
4	NITK-NLP_SUB1	73

TABLE VIII
Results for Magahi

Figure 1 shows the confusion matrix for SoMeWeTa.[16] Major problems arise for tags `ADJ` (15.5% recall), `ADV` (14.8%),

[14]We focus on recall; precision is very much the same as recall for all frequent labels, and higher for rare ones, since the taggers avoid predicting infrequent labels.

[15]One notable exception is that the BiLSTM tagger did non predict category `RD_ECH` at all (another hapax in the gold standard) but `RD_ECH_B` (once, incorrectly).

[16]Again, results are very similar for our other submissions.

PART (32.5%), and PROPN and X (both 0%), since these are quite frequent categories with abysmal error rates. As in the case of Bhojpuri, the tagger misclassifies them as NOUNs and VERB, which are the most frequent open classes. Moreover, the tagger frequently mistakes VERB for AUX (and AUX for VERB).

IV. Conclusion

The results for Bhojpuri are very satisfying. Close to 95% accuracy on a tagset of 33 tags with approximately 95,000 words of training data is also totally in line with what we would expect. It is a bit disappointing, though, that mindless parameter-tuning yields the best results – but the difference may very well not be significant.

The results for Magahi are very disappointing. Since we do not know the language, it is difficult for us to pinpoint the exact reasons for the bad performance, be it an over-generalization of our taggers, a shift in the tag distribution in the test data or an issue with the quality of the annotation. At least the use of additional resources outperforms mere parameter-tuning.

References

[1] Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. The Hindi/Urdu treebank project. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 659–697. Springer, 2017.

[2] Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[3] George Cardona. *The Indo-Aryan languages*, volume 9, pages 439–450. 15th edition, 1974.

[4] George Abraham Grierson. *Linguistic survey of India, Vol. V: Indo-Aryan Family, Eastern Group, Pt. II: Specimens of the Bihari and Oriya Languages*. Office of the Superintendent of Government Printing, India, Calcutta, 1903.

[5] Robert J. Jeffers. The position of the bihārī dialects in indo-aryan. *Indo-Iranian Journal*, 18(3):215–225, Nov 1976.

[6] Percy Liang. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2005.

[7] Office of the Registrar General & Census Commissioner. 2011 census data, Data on language and mother tongue, Statement 1: Abstract of speakers' strength of languages and mother tongues, 2011.

[8] Atul Ku Ojha, Pitambar Behera, Srishti Singh, and Girish N. Jha. Training & evaluation of POS taggers in Indo-Aryan languages: a case of Hindi, Odia and Bhojpuri. In *Proceedings of the 7th Language & Technology Conference (LTC 2015)*, pages 524–529, 2015.

[9] Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17, 2009.

[10] Thomas Proisl. SoMeWeTa: A part-of-speech tagger for German social media and web texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 665–670, Miyazaki, 2018. European Language Resources Association.

[11] Martin Riedl and Sebastian Padó. A named entity recognition shootout for German. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Volume 2: Short Papers*, pages 120–125, Melbourne, 2018.

[12] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259, Edmonton, 2003.

[13] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of EMNLP/VLC-2000*, pages 63–70, 2000.

[14] Manindra K. Verma. Bhojpuri. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 566–589. Routledge, London, 2003.

[15] Sheela Verma. Magahi. In George Cardona and Dhanesh Jain, editors, *The Indo-Aryan Languages*, pages 547–565. Routledge, London, 2003.