Analysing cross-lingual transfer in lemmatisation for Indian languages

Kumar Saunack*[1], Kumar Saurav*[1], and Pushpak Bhattacharyya[1]

[1]IIT Bombay
{krsaunack,krsrv,pb}@cse.iitb.ac.in

Abstract

Lemmatization aims to reduce the sparse data problem by relating the inflected forms of a word to its dictionary form. However, most of the prior work on this topic has focused on high resource languages. In this paper, we evaluate cross-lingual approaches for low resource languages, especially in the context of morphologically rich Indian languages. We test our model on six languages from two different families and develop linguistic insights into each model's performance.

1 Introduction

NLP has seen a sharp growth across various frontiers on multiple tasks; for example, today's systems are often required to generate text or to summarize documents. However, a morpheme remains the most basic level of information for most of them (Otter et al., 2020). Most of the research in these fields has focused on improving state-of-the-art for high resource languages. In contrast, research on low resource languages has been slow to start. For Indian languages, this is a major issue. Only some of the 22 scheduled Indian languages, which are a subset of the numerous languages spoken and written in India, have enough resources for training a deep learning model. For the remaining languages, the potential for improvement in performance is substantial.

Most of the current approaches for morphological analysis use cross-lingual transfer learning from a higher resource language to some low resource language (McCarthy et al., 2019). But choosing the high resource language for transfer learning is still done in an ad hoc manner, with the most common criteria being the phylogenetic distance in the language family (Cotterell and Heigold, 2017; Johnson et al., 2017). However, it has been shown that all languages from the same family might not share the same linguistic properties (Ahmad et al., 2019).

In this paper, we use different cross-lingual training methodologies and analyse the resulting source-target language pair performances based on different linguistic factors.

2 Models

We adapt the two-step attention process from the state of the art (Anastasopoulos and Neubig, 2019) on the SIGMORPHON 2019 morphological inflection task (McCarthy et al., 2019), switching the input and output to use it as a lemmatiser. The model has four parts: separate encoders for both the tags and the input character sequence, an attention mechanism, and a decoder.

The encoder for the lemma is single layer bidirectional LSTM. Morphological tags are also input to the model for which we use self-attention encoders (Vaswani et al., 2017) without

*These authors contributed equally to this work

positional embeddings, since the tag embeddings should be order-invariant. At each timestep, on the decoder side, two context vectors are created via two different attention matrices over the output from the encoding of lemma and tag (Luong et al., 2015).

The decoder then computes the output in a two-step process: it first creates a tag-informed state by attending over tags using the output from the decoder at the previous time step. We then compute the state vector by attending over the source characters using the tag-informed state. Using the updated state, the output character for that timestep is produced. This output is passed through a fully connected layer before applying a softmax to get the output character.

We also add structural bias to the attention model that encourages Markov assumption over alignments, that is, if the i-th source character is aligned to the j-th target one, alignments from the $(i+1)$-th or ith to $(j+1)$-th character are preferred.

We refer the reader to Cohn et al.(2016) for more details regarding the structural bias and Anastasopoulos and Neubig(2019) for more details and explanations about the two-step attention process.

3 Experiments

3.1 Data

From the SIGMORPHON 2019 shared task, we collect language data from the cross-lingual morphological inflection task for Bengali, Hindi, Kannada, Sanskrit, Telugu, and Urdu. Out of these, Telugu is the only language that does not have a large dataset. We use the same classification as the SIGMORPHON shared task for annotating a language as high or low resource.

A detailed description of the dataset that we use for training is provided in Table 1.

Language	Language Family	Script Type	Total	High	Low
Bengali (bn)	Indo-Aryan	LTR Abudgida	3,394	3,394	100
Hindi (hi)	Indo-Aryan	LTR Abudgida	10,000	10,000	100
Kannada (kn)	Dravidian	LTR Abudgida	3,506	3,506	100
Sanskrit (sa)	Indo-Aryan	LTR Abudgida	10,000	10,000	100
Telugu (te)	Dravidian	LTR Abudgida	61	-	61
Urdu (ur)	Indo-Aryan	RTL Abjad	10,000	10,000	100

Table 1: Number of inflected-word lemma pairs available for each language. The *Total* column shows the original number of samples and the *High* and *Low* columns show the curated training dataset size in a high and low resource setting respectively. During training, we augment the dataset to 10,000 samples in the low resource setting. *LTR*: Left-to-right, *RTL*: Right-to-left

We use the alignment method from Cotterell et al. (2016) to generate additional artificial data to augment the low resource datasets. The method relies on substituting multiple possible stems in a word with random sequences of characters while preserving its length.

For each language, the training data is augmented so that the total training set size is equal to 10,000, including the original training data.

3.2 Cross-lingual training

For the remainder of this section, let L_1 be the source language(high resource) and L_2 be the target language(low resource). We use a modified transfer learning method adapted from (Artetxe et al., 2020) that transfers learning from a model learnt on L_1 to another language L_2 based on results on a validation set (see Appendix B for more details).

The entire seq2seq model is broken up into modules, with the encoder, decoder, attention layers (called *EDA module* for the remainder of the section) the same for both source and transfer language. The embedding layers and the dense output layers are different for each language. The training then proceeds as follows (all the modules have been listed on the right side, with the trainable modules italicised):

	bn	hi	kn	sa	ur	average	mono
Bengali (bn)	-	**60**	59	57	59	58.75±1.09	58
Hindi (hi)	**45**	-	**45**	**45**	**45**	45.00±0.00	37
Kannada (kn)	52	**53**	-	44	48	49.25±3.56	49
Sanskrit (sa)	70	68	**74**	-	70	70.50±2.18	67
Telugu (te)	64	**82**	66	68	66	69.20±6.52	80
Urdu (ur)	**24**	23	20	10	-	19.25±5.54	26

(a) Our implementation

	bn	hi	kn	sa	ur	average	mono
Bengali (bn)	-	65	64	**67**	65	65.25±1.09	71
Hindi (hi)	40	-	**49**	46	43	44.50±3.35	34
Kannada (kn)	48	48	-	44	**50**	47.50±2.18	49
Sanskrit (sa)	64	**77**	60	-	66	66.75±6.30	59
Telugu (te)	**84**	78	**84**	80	80	81.20±2.40	72
Urdu (ur)	18	15	**19**	14	-	16.50±2.06	12

(b) Hard Monotonic Attention model

Table 2: Percentage accuracy of cross lingual models for different language pairs. The columns represent the high resource languages and the rows represent the low resource languages. *mono* column refers to the corresponding monolingual model.

Phase 1 - Copying phase for L_1 *EDA +L_1 embeddings +L_1 dense output*
The model is allowed to learn to copy characters. The copying phase is stopped when the accuracy reaches 80%. Attention heat maps after this phase show that the attention model has adapted to the structural biases and has learnt monotonicity.

Phase 2 Copying phase for L_2 *EDA +L_2 embeddings +L_2 dense output*
By learning to copy from L_2 accurately, we expect the embedding layer to learn proper representations of characters in L_2. This phase is stopped when the copying accuracy crosses 85%.

Phase 3 Training phase for L_1 *EDA +L_1* embeddings *+L_1 dense output*
L_1 embeddings weights are frozen and the model is allowed to train on the lemmatisation for high resource language. The model is expected to learn the process of lemmatisation.

Phase 4 Training phase for L_2 *EDA +L_2 embeddings +L_2 dense output*
We fine-tune the model on lemmatisation for L_2. We observe that the model converges quickly in this phase compared to Phase 3, although the time to convergence varies with different language pairs.

We use the model with the lowest validation loss for training the next phase in each case.

A total of 25 cross-lingual models are created. Since sufficient resources for Telugu were not available, models with Telugu as L_1 could not be created.

All the hyperparameters used are mentioned in Appendix A. We release all our code online for reproducibility and further research.

4 Results and Discussion

Table 2 lists the accuracy of our architecture and the hard monotonic attention model (Wu and Cotterell, 2019) for different language pairs in the context of cross-lingual as well as monolingual setting. The hard monotonic attention model in the cross-lingual setting was adapted from the SIGMORPHON 2019 shared task 1 (McCarthy et al., 2019).

4.1 Right to Left scripts

We see that both models achieve a very low accuracy for Urdu in the extremely low resource setting. Urdu as a source language in cross-lingual training is not effective as well - the accuracy values for the target languages lie within the corresponding standard deviation range.

To identify the possible source of low accuracy, we created models with reversed letter orders for Urdu, Hindi and Bengali. The accuracies do not change by much for both cross-lingual (with Urdu, Hindi, Bengali as target languages) and monolingual low-resource models. Therefore a right-to-left writing system is not the primary cause of low accuracy.

Therefore, we hypothesise that the Abjad script is more difficult to learn in a low resource setting because Abjad requires inferring vowels instead of explicitly supplying them. On running the models on Arabic, we obtain single-digit accuracy in all cases, which supports our claim.

4.2 Effect of source languages

Anastasopoulos et al. (2019) suspect that low variance in performance across source languages could be due to different scripts. We confirm the the hypothesis through our results here. There are 5 different scripts distributed among 6 languages in our dataset. For each transfer language and in each model, we can see that the deviation in performance is very small.

For example, we see that Bengali has a standard deviation of only around 1.09 in both the architectures, whereas the standard deviation for Sanskrit jumps to 6.30 for the hard attention model. The latter is due to the spike in performance when Hindi, a language very closely related to Sanskrit and using the same script, is used as a source language.

4.3 Performance gain over monolingual models

For a fixed transfer language, we can see that either almost all cross-lingual models perform better than the monolingual model or almost all cross-lingual models perform worse than the monolingual model, i.e., the performance of a few cross-lingual models can be generalised to all other source languages for a fixed transfer language. This fact is supported by the observation made in Section 4.2. Note that we compare the accuracy of cross-lingual and monolingual models for a given model architecture. For instance, Urdu consistently fares worse in our cross-lingual model, while it performs consistently better in the hard monotonic attention cross-lingual model. Note that we compare the gain/loss in performance against the monolingual model for that architecture.

Therefore, we claim that in extremely low resource settings, performance gains over monolingual models can be expected from all languages or languages closely related to the transfer language. The same result is observed for the morphological inflection task (Anastasopoulos and Neubig, 2019).

5 Related work

Lemmatisation has been tested extensively (Zeman et al., 2018; Nivre et al., 2017), but on datasets that are at least an order of magnitude greater than what we work with. Recently, there has been a shift to extremely low resource settings with the SIGMORPHON 2019 shared task (McCarthy et al., 2019) focusing on cross-lingual learning. However, their task focuses on the reverse direction: given a lemma and a bundle of morphological features, generate a target inflected form. To our knowledge, we are the first ones to study lemmatisation in such a low resource framework.

6 Conclusion

Inference-based scripts such as Abjad can be difficult for models to learn in extremely low resource scenarios. For other scripts, it is difficult to predict whether cross-lingual models fare better than monolingual models. In general, for a given low resource language, the performance of a language as a source language is a good predictor of gain/loss for other source languages.

Acknowledgements

We are grateful to the members of the CFILT lab at IIT Bombay for providing us the resources required to finish this project. We are also thankful to Kalpesh Krishna, a Ph.D. candidate at University of Massachussets, Amherst for help where required.

References

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. *arXiv preprint arXiv:1908.05838*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online, July. Association for Computational Linguistics.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 876–885, San Diego, California, June. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark, September. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy, August. Association for Computational Linguistics.

Joakim Nivre, Lars Ahrenberg ˇZeljko Agic, et al. 2017. Universal dependencies 2.0–conll 2017 shared task development and test data. *LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.*

Daniel W Otter, Julian R Medina, and Jugal K Kalita. 2020. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Shijie Wu and Ryan Cotterell. 2019. Exact hard monotonic attention for character-level transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1530–1537, Florence, Italy, July. Association for Computational Linguistics.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.

A Hyperparameters

All our models were trained on a single 12 GB Nvidia GeForceGTX TitanXGPU. We use the Adam optimiser with the default parameters except for learning rate. The training time for each model was between 1 to 3 hours.

Note that the cross-lingual method that we use corresponds to the method described by Artexte et al. (2019) and so there are 4 phases of training. We list out the hypermaters as comma separated values:

- Batch size: 10
- Training epochs: 10,10,10,10
- Activation function: Swish
- Learning rate: 10e-3,10e-3,10e-3,10e-3

B Validating cross-lingual training method

Full	Embedding	Encoder,Decoder, Attention	FCN/ Dense	Best accuracy
P1,P2,P4	-	P3	P3	**60**
P1,P4	P2	P3	P2,P3	58
P1	P2,P4	P3	P2,P3,P4	34
P1,P2	-	P3,P4	P3,P4	57
P1	P2	P3,P4	P2,P3,P4	59

Table 3: Validation accuracy on Hindi-Bengali cross-lingual models when different different parts of the model are frozen in different phrases(Sec **??**). P3 and P1 (training phase for higher resource language) remain unchanged in all rows. Each column represents the trainable part of the model

The table shows the accuracy when we freeze different parts of the model during different phases of training. The resulting models are evaluated on a validation set. We choose the model with the best accuracy as the model of our choice.

C Monolingual model

We also trained monolingual models for comparison. We list out the hyperparameters that we use for training them:

- Batch size: 10
- Training epochs: 10
- Activation function: Swish
- Learning rate: 10e-3