Project#21: A Dependency Parser for Hindi with Zero-Shot Cross-Lingual Transfer Learning for Marathi

Abhishek Jaiswal[1], **A.V.S.D.S.Mahesh**[2], **Tushar Shandhilya**[3]
[1]19111262, [2]19111265, [3]19111418
[1]CSE, [2]CSE, [3]CSE
{abhijais, maheshak, stushar}@cse.iitk.ac.in

Abstract

We have built a dependency parser for Hindi with a web-interface and, further demonstrated its cross-lingual usage by applying on Marathi, a language that is low-resourced yet not very different from the former. For our parser, we have applied and tested various techniques ranging from transition-based parser that use SVM as a classifier to deep neural network based parsers which incorporate recent context-based word embeddings like BERT and FastText. We have demonstrated the immediate zero-shot application of such a parser on Marathi and also applied an existing word-embedding alignment method called MUSE to improve the cross-lingual application performance. In the future, we aim to apply these techniques to Bhojpuri, Telugu and Tamil.

1 Introduction

Dependency parsing is a syntactic parse of a sentence described by binary relations (called dependency relations) between the words present in the sentence. It does not involve phrase structure rules which generate a parse tree for a given sentence. Importantly, dependency parsing allows one to work with language that would have free word order. Indian languages, though prefer an order, are necessarily free word order. Thus dependency parsing is useful in this scenario. A dependency parse of a sentence is useful in higher tasks like information retrieval, question and answering, co-reference resolution etc.(Jurafsky and Martin, 2009) In this work, we have applied various dependency parsing techniques of Hindi, a most widely spoken Indian language and use the parser and apply it on a yet another widely spoken(nevertheless, under-resourced) Indian language, Marathi in a cross-lingual zero-shot transfer learning scenario. In the future, we aim to extend this to Bhojpuri, which is even under-resourced than Marathi.

Various methods have been applied till date on Hindi dependency parsing ranging from traditional methods as in (Nivre, 2009),(Jain et al., 2012) to neural-network based methods as in (Dozat et al., 2017) and (Bhat et al., 2018) which achieve considerably good performance with best 94.70% UAS(Unlabeled Attachment Score) and 91.59% LAS(Labeled Attachment Score) reported in (Dozat et al., 2017). On the other hand, (Tandon and Sharma, 2017) presented significant results on Marathi i.e. 85.98% UAS and 69.01% LAS, despite the fact it is low-resourced. Nevertheless, this is not still up to the level of Hindi. There are many other Indian languages that lack a sufficient amount of data itself for training. Thus it is desirable to have zero-shot or few-shot learning methods to overcome the problem of data scarcity. Some of the existing methods in this direction are (Schuster et al., 2019),(Wang et al., 2019), (Tran and Bisazza, 2019) and (Kondratyuk, 2019). On Zero-Shot learning for Marathi, (Kondratyuk, 2019) reports performance of 79.37% UAS and 67.72% LAS. None have been observed to the best of our knowledge on Bhojpuri. Application of zero-shot learning on Telugu and Tamil have been found in (Tran and Bisazza, 2019) and (Kondratyuk, 2019).

In the rest of this paper, we shall first briefly mention the approaches of dependency parsing. Later we shall present a brief overview of work related to dependency parsing approaches used for Hindi and other Indian Languages. Then we shall talk about approaches we have applied in our work followed by discussions on corpus considered, experimentation, results and finally conclude.

2 Problem Defintion

A dependency parser takes a sentence as input and outputs a syntactic structure called a *dependency graph* $G(V, E)$ with lexical nodes(words) as ver-

tices V and binary relations or *dependencies* as edges(directional) E. Each edge or a dependency $(i, j, l) \in E$ is defined from a word i called as *head* to another word j called as a *modifier* with *dependency label* l. The labels may be not present as well. In the latter case, the dependency graph is called *unlabeled*. It is required that a dependency graph be acyclic thus is a tree(Nivre, 2003). An example of a dependency graph is shown in figure 1.

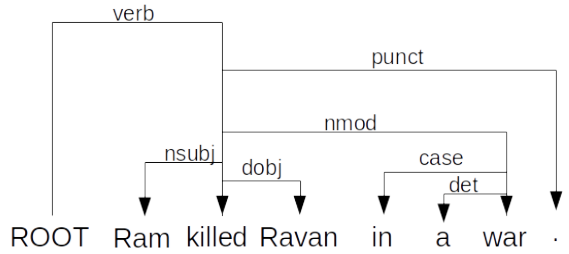

Figure 1: Example of a dependency parse

There are two main approaches in arriving at a dependency parse namely graph-based and transition based. In the graph-based approach every edge (i, j, l) is associated with a score $s(i, j, l)$ and the problem is finding a spanning-tree with maximum score where score of a graph is the sum of scores of individual edges. The problem can be written as:

$$G = \arg\max_{G(V,E)} \sum_{(i,j,l)\in E} s(i, j, l)$$

Transition based parsers work in a way similar to shift-reduce parsers which apply transitions T on an abstract machine taking it from an initial configuration to a final configuration while constructing a dependency graph. Each transition t is associated with a score $s(c, t)$ where c is the configuration at which the transition occurs. The next transitiont^* to be taken is the one with the maximum score:

$$t^* = \arg\max_{t\in T} s(c, t)$$

The key problem lies in coming up with proper feature representations of edges and configurations respectively in graph-based and transition-based approaches and their scoring functions. Given these, graph-based approaches work in O(n) time while transition-based approaches work in O(n^2).(Nivre and McDonald, 2008)

3 Related Work

There is much literature present on dependency parsing in general, we shall confine to that which is relevant or rather applied or shall be applied in our work. The first approach we applied used transition based Arc-eager parser (Nivre, 2008) which uses an SVM classifier to classify the next transition. Similar approach was used in MaltParser (Nivre et al., 2007), which was applied on Hindi (Nivre, 2009) to get performance of 89.4% UAS and 78.2% LAS. This was further tweaked in (Jain et al., 2012) to achieve a performance of 91.7% UAS and 83.9% LAS on Hindi.

Usage of neural networks(NN) to classify the transitions was first introduced in (Chen and Manning, 2014). Usage of a bidirectional LSTM(BiLSTM) to encode the input features and further classification (in case of transition-based) and regression(scoring in case of graph-based) using NN is found in (Kiperwasser and Goldberg, 2016), one of the architecture used in this work. The NN used for scoring in the graph-based approach is incorporated with biaffine attention in (Dozat and Manning, 2016) which was applied in (Dozat et al., 2017) to get the state-of-the-art results of 94.7% UAS and 91.6% LAS on Hindi. Context-based word embeddings called ElMo (Peters et al., 2018) are used in (Schuster et al., 2019) as input embeddings and further alignment of word embeddings of the source language to that of the target language is achieved through MUSE(Conneau et al., 2017) which is fed into (Dozat and Manning, 2016) network to achieve zero-shot cross-lingual learning in the same work. For this instead of ElMo we have used BERT(Devlin et al., 2018), the existing state-of-the-art context based embeddings, trained for multiple-languages including Hindi and Marathi. We have lately realized that BERT has been already used in (Kondratyuk, 2019) to achieve zero-shot state-of-the-art performance for many languages with Marathi one of them at performance of 79.37% UAS and 67.72% LAS.

4 Proposed Approach

The neural network model we considered (Kiperwasser and Goldberg, 2016) uses deep Bi-directional LSTMs to obtain word encodings as a feature that is given to the parser. The training is performed along with the parsing and the error is back-propagated. The input to our model will be a sentence in our target language and the output will

be its dependency parse information such as lemma, POS tag, parent word, etc. To handle cross-lingual transfer, we use the method presented in (Schuster et al., 2019) by obtaining contextual embedding of words for both source and target languages and aligning them by using a bilingual dictionary. The authors claim that it helps in increasing the performance but can do well without a dictionary in an unsupervised approach using a discriminator to train the network.

We have used the graph-based parsing as given in (Kiperwasser and Goldberg, 2016) which uses the arc-factored graph as presented in (McDonald et al., 2005). For a sentence x, Arc-factored parsing decomposes the score of a tree y which belongs to all possible parse trees Y to the sum of the score s of its head-modifier edge(i, j):

$$parse = \arg\max_{y \in Y} \sum_{(h,m) \in y} s(\phi(i, j))$$

The feature extractor uses the concatenation of head and modifier embeddings:

$$\phi(i, j) = BIRNN(x, i) o BIRNN(x, j)$$

Finally, score s is obtained through the multi-layer perceptron MLP. This architecture is summarized in figure 2.

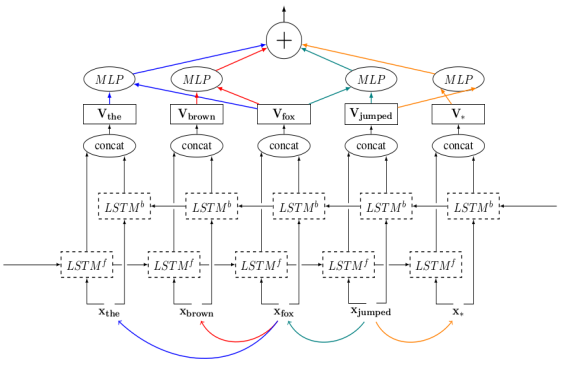

Figure 2: The neural model discussed in (Kiperwasser and Goldberg, 2016)

The word embeddings are aligned for cross-lingual learning in the following way. Given an embedding $e^s_{i,c}$ in source language s, we want to generate an embedding $e^{s->t}_{i,c}$ in the target language t space, using a linear mapping $W^{s->t}$ given by

$$e^{s->t}_{i,c} = W^{s->t} e^s_{i,c}$$

.

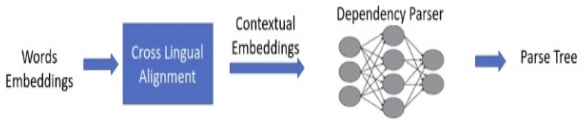

Figure 3: Pipeline

We have trained the dependency parser on the aligned embeddings of Hindi and used the trained model for zero-shot cross lingual learning on Marathi. The overall pipeline is presented in figure 3.

5 Corpus/Data Description

In this work, we are primarily using two types of data. One is in CONLLU format which contains information of lemma, POS tag, parent word, relation, etc i.e. the tree-banks which are provided by the Universal dependencies (Zeman et al., 2019) for Hindi, (Bhat et al.; Palmer et al., 2009) and Marathi (Ravishankar, 2017). The Second type of data is Wikipedia dump. The different statistics for the data provided in tables 1 and 2.

Language	Sentences	Tokens
Hindi	16,647	351,704
Bhojpuri	254	4,881
Marathi	466	3,506

Table 1: Universal Dependencies data statistics

Language	Sentences	Tokens
Hindi	1,743,340	1,085,965
Marathi	397,185	618,773

Table 2: Wikipedia dump data statistics

6 Experiments and Results

Firstly, we have aligned (using MUSE with BERT embeddings) Hindi to Marathi space and trained on this new Hindi embeddings on deep biaffine parser. This model was tested on Hindi and Marathi test sets. We trained three types of model: without external embedding, with Hindi aligned embeddings, and with both Hindi and Marathi embeddings as input.

Then we also tested a bidirectional LSTM parser with the above cases as the biaffine parser was internally using multilingual BERT. We have also

got access to pre-trained FastText embeddings and hence the above tests were repeated with FastText input embeddings on BiLSTM parser.

All the experimental results have been summarised in the Table 3 and the demo can be found here. [1]

Most of the initial results with MUSE based embeddings are not very promising because they were trained with a very small dictionary and hence the alignments produced were not of good quality. Comparatively, pretrained Fasttext embeddings give better results because they are trained on a larger corpus.

Method	**LAS(%)**	**UAS(%)**
using Bert based embeddings		
Hindi(Arc-Eager)	79	89
Hindi(Kipperwasser,16)	89	91
Hindi(Dozat17)	**89**	**93**
Marathi(Arc-Eager)	**61**	**75**
Marathi after alignment (Kipperwasser,16)	45	59
Marathi without alignment (Kipperwasser,16)	50	67
Marathi after alignment (Dozat17)	45.7	60.2
Marathi without alignment (Dozat17) (With alignment)	46.6	60.8
using Fasttext embeddings		
Hindi(Kipperwasser,16)	**89**	**93**
Marathi after alignment (Kipperwasser,16)	48	64

Table 3: Results Comparison

7 Error Analysis

The results for Hindi almost match that of the state of the art but only about a percent less. This should only be because our models are not well-tuned which would some extra training time. The same holds in the standalone training of Marathi i.e. not going for zero-shot. In the case of zero-shot, results even without alignment are surprisingly good attributing to the mystery behind multi-lingual BERT. On the other hand, results, after alignment, were not up to the mark in both cases, because MUSE alignments were not very good since they were obtained using a small dictionary and working on a better dictionary should improve these results.

[1]Demo link: maheshak.cse.iitk.ac.in (only accessible in IITK's local network)

8 Individual Contribution

The contributions of each team member is summarized in table 4.

Name	Contribution
Mahesh	Applied arc-eager parser (from nltk) on Hindi and Marathi, developed the back-end for the website application and Wikipedia dump pre-processor(not used till now) and performed literature survey.
Tushar	Built a webapp for the parser and fine-tuned BERT to get Hindi and Marathi embeddings. Trained unsupervised MUSE to get aligned embeddings and trained those on biaffine parser. Also scraped Hindi and Marathi language data from Wikipedia
Abhishek	Built Hindi-Marathi Bilingual dictionary and a POS tagger for statistical parser. Trained supervised and unsupervised embedding alignments using MUSE for Hindi-Marathi and Marathi-Hindi with Bert and Fasttext. Trained BiLSTM and biaffine parsers and compared results for the above cases.

Table 4: Our Contributions

9 Future Work

As we have lately found out regarding the already existing state-of-the-art approaches, we would first like to keep up to date with them. Secondly, due to lack of time, we have used already existing codes and pre-trained models thus could not have much control and as a result, we would like to code at a finer level and train the word embeddings ourselves. Next, we would like to apply these methods on Bhojpuri, Telugu and Tamil. Also half of the sentences in Marathi consist of sandhi which needs to be considered.

10 Conclusion

We have thus built a Hindi dependency Parser with a web interface with comparable performance to recent ones. We have also applied existing methods for zero-shot learning for Marathi and have thus built a basic setup for further improvement and further applications on different languages.

11 Presentation Feedback

During the presentation, questions were raised regarding why certain layers were used in the parsing model architecture, specifically why summing the scores of different edges but not a neural network on top. As the scoring was part of the MST algorithm, there is no point in using a neural network to learn a well established already existing deterministic algorithm. Thus this question was closed there itself.

Also, we were asked why we choose Marathi as our target language and this is due to two things, availability of Marathi data at universal treebanks and secondly, it is pretrained with BERT. In the case of Bhojpuri, the former is true but not the latter. In the case of Bengali, the latter is true but not the former. Taking into consideration the amount of time available we had to go with Marathi. Nevertheless, in the future, we shall train word embeddings for Bhojpuri as well.

We received some very valuable suggestions like we were suggested by Karthikeyan to look at VecMap [2] to learn cross lingual word embeddings instead of MUSE and to look at LangRank [3] to compare languages in terms of their similarity for best transfer learning task. We have also received suggestions from Rahul to consider Dravidian languages and indeed that is mentioned as a part of our future work. We have also been suggested by Major Prateek to include the devanagari input method in the web interface and we shall be implementing it soon in the future.

References

Irshad Bhat, Riyaz A. Bhat, Manish Shrivastava, and Dipti Sharma. 2018. Universal dependency parsing for Hindi-English code-switching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 987–998, New Orleans, Louisiana. Association for Computational Linguistics.

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 *Handbook of Linguistic Annotation*. Springer Press.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

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.

Naman Jain, Karan Singla, Aniruddha Tammewar, and Sambhav Jain. 2012. Two-stage approach for Hindi dependency parsing using MaltParser. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*, pages 163–170, Mumbai, India. The COLING 2012 Organizing Committee.

Dan Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognintion, Second Edition*. Pearson Education Inc.

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

Daniel Kondratyuk. 2019. 75 languages, 1 model: Parsing universal dependencies universally. *CoRR*, abs/1904.02099.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd*

[2] VecMap, https://github.com/artetxem/vecmap

[3] LangRank, https://github.com/neulab/langrank

Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 91–98, Ann Arbor, Michigan. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34:513–553.

Joakim Nivre. 2009. Parsing indian languages with maltparser. *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pages 12–18.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio. Association for Computational Linguistics.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Ravishankar. 2017. Marathi Dependency Tree data. https://github.com/UniversalDependencies/UD_Marathi-UFAL.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Juhi Tandon and Dipti Misra Sharma. 2017. Unity in diversity: A unified parsing strategy for major Indian languages. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 255–265, Pisa,Italy. Linköping University Electronic Press.

Ke Tran and Arianna Bisazza. 2019. Zero-shot dependency parsing with pre-trained multilingual sentence representations. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 281–288, Hong Kong, China. Association for Computational Linguistics.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, Hong Kong, China. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, and Abrams et al. 2019. Universal dependencies 2.5. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.