Cross-lingual transfer parsing from Hindi to Bengali using delexicalization and chunking

Ayan Das, Agnivo Saha, Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur, WB, India
ayan.das@cse.iitkgp.ernet.in
agnivo.saha@gmail.com
sudeshna@cse.iitkgp.ernet.in

Abstract

While statistical methods have been very effective in developing NLP tools, the use of linguistic tools and understanding of language structure can make these tools better. Cross-lingual parser construction has been used to develop parsers for languages with no annotated treebank. Delexicalized parsers that use only POS tags can be transferred to a new target language. But the success of a delexicalized transfer parser depends on the syntactic closeness between the source and target languages. The understanding of the linguistic similarities and differences between the languages can be used to improve the parser. In this paper, we use a method based on cross-lingual model transfer to transfer a Hindi parser to Bengali. The technique does not need any parallel corpora but makes use of chunkers of these languages. We observe that while the two languages share broad similarities, Bengali and Hindi phrases do not have identical construction. We can improve the transfer based parser if the parser is transferred at the chunk level. Based on this we present a method to use chunkers to develop a cross-lingual parser for Bengali which results in an improvement of unlabelled attachment score (UAS) from 65.1 (baseline parser) to 78.2.

1 Introduction

Parsers have a very important role in various natural language processing tasks. Machine learning based methods are most commonly used for learning parsers for a language given annotated parse trees which are called treebanks. But treebanks are not available for all languages, or only small treebanks may be available. In recent years, considerable efforts have been put to develop dependency parsers for low-resource languages. In the absence of treebank for a language, there has been research in using cross-lingual parsing methods (McDonald et al., 2011) where a treebank from a related source language (SL), is used to develop a parser for a target language (TL). In such work, an annotated treebank in SL and other resources in are used to develop a parser model for TL. Most of the existing work assume that although annotated treebanks are not available for the target language TL, there are other resources available such as parallel corpus between the source and the target languages (Xiao and Guo, 2015; Rasooli and Collins, 2015; Tiedemann, 2015). However, developing a parallel corpus is also expensive if such parallel corpus is not available.

In this work, our goal is to look at methods for developing a cross-lingual transfer parser for resource poor Indian language for which we have access to a small or no treebank. We assume the availability of a monolingual corpus in target language and a small bilingual (source-target) dictionary.

Given our familiarity with Bengali and Hindi, and availability of a small treebank we aim to test our approach in Hindi-Bengali transfer parsing. We choose Hindi as the source language as it is syntactically related to Bengali and a Hindi treebank (Nivre et al., 2016) is freely available which can be used to train a reasonably accurate parser (Saha and Sarkar, 2016). We wish to use this Hindi treebank to develop a Bengali parser. Although our current work aims to develop a parser in Bengali from Hindi, this may be taken up as a general method for other resource poor languages. We also have access to a monolingual corpus in Bengali and a small bilingual (Hindi-

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 99–108,
Varanasi, India. December 2016.

Bengali) dictionary.

Since the vocabulary of two languages are different, some of the work in the literature attempted to address this problem by delexicalizing the dependency parsers by replacing the language-specific word-level features by more general part-of-speech or POS-level features. Such methods have yielded moderate quality parsers in the target language (McDonald et al., 2011). However the number of POS features is small and may not contain enough information. In order to alleviate this problem some work have been proposed to incorporate word-level features in the form of bi-lingual word clusters (Täckström et al., 2012) and other bilingual word features (Durrett et al., 2012; Xiao and Guo, 2014).

Both Hindi and Bengali use the SOV (Subject-Object-Verb) sentence structure. However, there exist differences in the morphological structure of words and phrases between these two languages (Chatterji et al., 2014). Since the overall syntactic structure of the languages are similar, we hypothesize that chunk level transfer of a Hindi parser to Bengali may be more helpful than word-level transfer.

The rest of the paper is organized as follows. Section 2 discusses some of the existing related work. In Section 3 we state the objective of this work. In Section 4 we present in details the the dataset used, and in 5 we state in details our approach for cross-lingual parsing. In Section 6 we analyze the errors. Section 7 concludes the paper.

2 Related work

A variety of methods for developing transfer parsers for resource poor languages without any treebank have been proposed in the literature. In this section, we provide a brief survey of some of the methods relevant to our work.

2.1 Delexicalized parsing

Delexicalized parsing proposed by Zeman and Resnik (2008) involves training a parser model on a treebank of a resource-rich language in a supervised manner without using any lexical features and applying the model directly to parse sentences in target language. They built a Swedish dependency parser using Danish, a syntactically similar language. Søgaard (2011) used a similar method for several different language pairs. Their system performance varied widely (F1-score : 50%-75%) depending upon the similarity of the language pairs.

Täckström et al. (2012) used cross-lingual word clusters obtained from a large unlabelled corpora as additional features in their delexicalized parser. Naseem et al. (2012) proposed a method for multilingual learning to languages that exhibit significant differences from existing resource-rich languages which selectively learns the features relevant for a target language and ties the model parameters accordingly. Täckström et al. (2013) improved performance of delexicalized parser by incorporating selective sharing of model parameters based on typological information into a discriminative graph-based parser model.

Distributed representation of words (Mikolov et al., 2013b) as vector can be used to capture cross-lingual lexical information and can be augmented with delexicalized parsers. Xiao and Guo (2014) learnt language-independent word representations to address cross-lingual dependency parsing. They combined all sentences from both languages to induce real-valued distributed representation of words under a deep neural network architecture, and then use the induced interlingual word representation as augmenting features to train a delexicalized dependency parser. Duong et al. (2015a) followed a similar approach where the vectors for both the languages are learnt using a skipgram-like method in which the system was trained to predict the POS tags of the context words instead of the words themselves.

2.2 Cross-lingual projection

Cross-lingual projection based approaches use parallel data or some other lexical resource such as dictionary to project source language dependency relations to target language (Hwa et al., 2005). Ganchev et al. (2009) used generative and discriminative models for dependency grammar induction that use word-level alignments and a source language parser.

McDonald et al. (2011) learnt a delexicalized parser in English language and then used the English parser to seed a constraint learning algorithm to learn a parser in the target language. Ma and Xia (2014) used word alignments obtained from parallel data to transfer source language constraints to the target side.

Rasooli and Collins (2015) proposed a method to induce dependency parser in the target language

using a dependency parser in the source language and a parallel corpus. Guo et al. (2015) proposed a CCA based projection method and a projection method based on word alignments obtained from parallel corpus.

2.3 Parsing in Hindi and Bengali

Hindi and Bengali are morphologically rich and relaively free word order languages. Some of the notable works on Indian languages are by Bharati and Sangal (1993) and Bharati et al. (2002). Also the works of Nivre (2005) and Nivre (2009) have been successfully applied for parsing Indian languages such as Hindi and Bengali. Several works on Hindi parsing (Ambati et al., 2010; Kosaraju et al., 2010) used data-driven parsers such as the Malt parser (Nivre, 2005) and the MST parser (Mcdonald et al., 2005). Bharati et al. (2009b) used a demand-frame based approach for Hindi parsing. Chatterji et al. (2009) have shown that proper feature selection (Begum et al., 2011) can immensely improve the performance of the data-driven and frame-based parsers.

Chunking (shallow parsing) has been used successfully to develop good quality parsers in Hindi language (Bharati et al., 2009b; Chatterji et al., 2012). Bharati et al. (2009b) have proposed a two-stage constraint-based approach where they first tried to extract the intra-chunk dependencies and resolve the inter-chunk dependencies in the second stage. Ambati et al. (2010) used disjoint sets dependency relation and performed the intra-chunk parsing and inter-chunk parsing separately. Chatterji et al. (2012) proposed a three stage approach where a rule-based inter-chunk parsing followed a data-driven inter-chunk parsing.

A project for building multi-representational and multi-layered treebanks for Hindi and Urdu (Bhatt et al., 2009)[1] was carried out as a joint effort by IIIT Hyderabad, University of Colorado and University of Washington. Besides the syntactic version of the treebank being developed by IIIT Hyderabad (Ambati et al., 2011), University of Colorado has built the Hindi-Urdu proposition bank (Vaidya et al., 2014) and a phrase-structure form of the treebank (Bhatt and Xia, 2012) is being developed at University of Washington. A part of the Hindi dependency treebank[2] has been released in which the inter-chunk dependency relations (dependency links between chunk heads) have been manually tagged and the chunks were expanded automatically using an arc-eager algorithm.

Some of the major works on parsing in Bengali language appeared in ICON 2009 (http://www.icon2009.in/). Ghosh et al. (2009) used a CRF based hybrid method, Chatterji et al. (2009) used variations of the transition based dependency parsing. Mannem (2009) came up with a bi-directional incremental parsing and perceptron learning approach and De et al. (2009) used a constraint-based method. Das et al. (2012) compares performance of a grammar driven parser and a modified MALT parser.

3 Objective

We want to build a good dependency parser using cross-lingual transfer method for some Indian languages for which no treebanks are available. We try to make use of the Hindi treebank to build the dependency parser. We explore the use of the other resources that we have.

Due to our familiarity with Bengali language and availability of a small treebank in Bengali we aim to perform our initial experiments in Bengali to test our proposed method. We have a small Hindi-Bengali bilingual dictionary and POS taggers, morphological analyzers and chunkers for both these languages.

In such a scenario delexicalization methods can be used for cross-lingual parser construction. We wish to get some understanding of what additional resources can be used for general cross-lingual transfer parsing in this framework depending on the similarity and differences between the language pairs.

4 Resources used

For our experiments, we used the Hindi Universal Dependency treebank to train the Hindi parser (Saha and Sarkar, 2016; Chen and Manning, 2014). The Hindi universal treebank consists of 16648 parse trees annotated using Universal Dependency (UD) tagset divided into training, development and test sets. For testing in Bengali we used the test set of 150 parse trees annotated using Anncorra (Sharma et al., 2007) tagset. This small Bengali treebank was used in ICON2010[3]

[1] http://verbs.colorado.edu/hindiurdu/index.html
[2] http://ltrc.iiit.ac.in/treebank_H2014/
[3] http://www.icon2010.in/

contest to train parsers for various Indian langauges. The parse trees in the test data were partially tagged with only inter-chunk dependencies and chunk information. We completed the trees by manually annotating the intra-chunk dependencies using the intra-chunk tags proposed by Kosaraju et al. (2012). We used the complete trees for our experiments.

Table 1 gives the details of the datasets used.

Table 1: Universal Dependency Hindi treebank.

Data	Universal Dependency treebank (Number of trees)	ICON Bengali treebank (Number of trees)
Training	13304	979
Development	1659	150
Test	1685	150

The initial Hindi and Bengali word embeddings were obtained by running word2vec (Mikolov et al., 2013b) on Hindi Wikipedia dump corpus and FIRE 2011[4] corpus respectively.

For Hindi-Bengali word pairs we used a small bilingual dictionary developed at our institute as a part of ILMT project[5]. It consists of about 12500 entries. For chunking we used the chunkers and chunk-head computation tool developed at our institute. The sentences in the Hindi treebank were chunked using an automatic chunker to obtain the chunk-level features. In case of disagreement between the output of automatic chunker and the gold standard parse trees we adhered to the chunk structure of the gold standard parse tree.

Before parsing the Hindi trees we relabeled the Hindi treebank sentences by Anncorra (Sharma et al., 2007) POS and morphological tags using the POS tagger (Dandapat et al., 2004) and morphological analyzer (Bhattacharya et al., 2005) as the automatic chunker requires the POS and morphological information in Anncorra format. Moreover, due to relabeling both the training and the test data will have the POS and morphological features in Anncorra format.

5 Our proposed Hindi to Bengali cross-lingual dependency parser

5.1 Baseline delexicalization based method

For the delexicalized baseline we trained the Hindi parser using only POS features. We used this model directly to parse the Bengali test sentences. It gives an UAS (Unlabelled Attachment Score) of 65.1% (Table 2).

We report only the UAS because the Bengali arc labels uses AnnCorra tagset which is different from Universal Dependency tagset. The dependency lables in the UD and ICON treebanks are different, with ICON providing a more fine-grained and Indian language specific tags. However, it was observed that the unlabelled dependencies were sufficiently similar.

5.2 Transferred parser enhanced with lexical features

When the parser trained using the lexical features of one language is used to parse sentences in another language the performance depends on the lexical similarity between the two languages.

We wish to investigate whether it is possible to use the syntactic similarities of the words to transfer some information to the Bengali parser along with the non-lexical information. We have used word embeddings (Mikolov et al., 2013b) for the lexical features in the hope that the word vectors capture sufficient lexical information.

Our work is different from that of (Xiao and Guo, 2014) and (Duong et al., 2015b) where the word vectors for both the languages are jointly trained. We observed that the work of (Xiao and Guo, 2014) is dependent on the quality and size of the dictionary and the training may not be uniform due to the difference in frequency of the words occurring in the corpus on which the vectors are trained. It also misses out the words that have multiple meanings in the other language.

Our method has the following steps;

Step 1 - Learning monolingual word embeddings : The monolingual word embeddings for Hindi and Bengali are learnt by training word2vec (Mikolov et al., 2013b) on monolingual Hindi and Bengali corpus respectively. The dimension of the learnt word embeddings are set to 50.

Step 2 - Training the Hindi monolingual dependency parser : To train the Hindi parser model using the Hindi treebank data we used the parser proposed by Chen and Manning (2014). The word embeddings were initialized by the ones learnt from monolingual corpus. Apart from the word embeddings, the other features are randomly initialized.

[4]http://www.isical.ac.in/ clia/2011/

[5]http://ilmt.iiit.ac.in/ilmt/index.php

Step 3 - Learning interlingual word representations using linear regression based projection: For learning interlingual word representations we used all the cross-lingual word pairs from a Hindi-Bengali dictionary and dropped the Hindi words whose corresponding entry in Bengali is of multiple words. We used only those word pairs for which both the words are in the vocabulary of the corresponding monolingual corpora on which the word embeddings were trained. The linear regression method (Mikolov et al., 2013a) was used to project the Bengali word embeddings into the vector space of the Hindi embeddings obtained after training the parser on Hindi treebank data. The regressor was trained using the embeddings of the 3758 word-pairs obtained from the dictionary.

Subsequently, we attempted to compare the method proposed by Xiao and Guo (2014). In both the cases the parser performances were very similar and hence we report only the results obtained using linear regression.

Step 4 - Transfer of parser model from Hindi to Bengali : In the delexicalized version, the parsers are used directly to test on Bengali data. In the lexicalized versions, we obtained the Bengali parser models by replacing the Hindi word embeddings by the projected Bengali word vectors obtained in Step 3. The transformation is shown in figure 1.

Table 2: Comparison of 1) delexicalized parser model and 2) parser using projected Bengali vectors.

	Delexicalized (Baseline)	Projected Bengali vectors
(Chen and Manning, 2014) parser	65.1	67.2

Table 2 compares the UAS of word-level transfer for the 1) delexicalized parser model (*Delexicalized*) and 2) the lexicalized Bengali parser model in which the Hindi word embeddings are replaced by Bengali word vectors projected onto the vector space of the Hindi word embeddings (*Projected Bengali vectors*). We observe that projected lexical features improves UAS over the delexicalized baseline from 65.1 to 67.2.

5.3 Chunk-level transfer for cross-lingual parsing

There exist differences in the morphological structure of words and phrases between Hindi and Bengali. For example, the English phrase "took bath" is written in Hindi as "*nahayA*" using a single word and the same phrase in Bengali is written as "*snan korlo*" "(bath did)" using two words. Similarly, the English phrase "is going" is written in Hindi as "*ja raha hai*" "(go doing is)" using three words and the same phrase in Bengali is written as "*jachhe*" using a single word.

This makes us believe that chunking can help to improve cross-lingual parsing between Hindi and Bengali languages by using the similarities in the arrangement of phrases in a sentence. Chunking (shallow parsing) reduces the complexity of full parsing by identifying non-recursive cores of different types of phrases in the text (Peh and Ann, 1996). Chunking is easier than parsing and both rule-based chunker or statistical chunker can be developed quite easily.

In figure 2 we present a Bengali sentence and the corresponding Hindi sentence. They are transliterated to Roman. The English gloss of the sentences are given. We indicate by parentheses the chunks of the sentences. We indicate by line the correspondence between the chunks. We see that the correspondence is at the chunk level and not at the word level.

The sentences are quite similar as far as the inter-chunk orientation is concerned as is evident from the Figure 3 and 4.

We have used Hindi and Bengali chunkers which identify the chunks and assign each chunk to its chunk type, chunk-level morphological features and the head words. For chunk level transfer we performed the following steps:

Step 1: We chunked the Hindi treebank sentences and extracted the chunk heads.

Step 2: We converted the full trees to chunk head trees by removing the non-head words and their links such that only the chunk head words and their links with the other head words are left.

Step 3: We trained the Hindi dependency parsers using the Hindi chunk head trees by the delexicalization method and the method described in section 5.2.

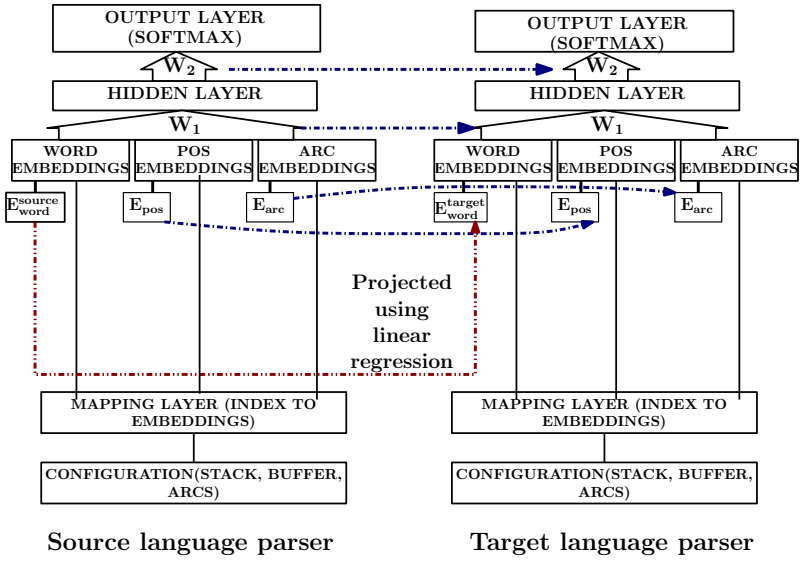

Figure 1: The neural network shares parameters like weights and POS, arc-label embeddings in source and target language parser models. Only the source language word embeddings replaced by projected target language word vectors. $E^{source}_{word}, E^{target}_{word}, E_{POS}, E_{arc}$ are the embedding matrices from which the mapping layer gets the vectors by indexing.

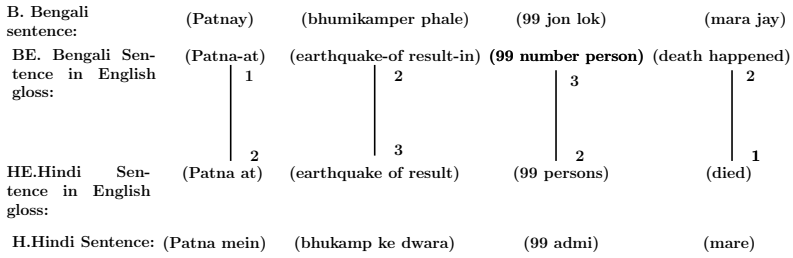

Figure 2: Chunk mapping between a Bengali and Hindi sentence that conveys the same meaning : "99 people died due to earthquake in Patna".

Figure 3: Word-level parse trees of the example Bengali and Hindi sentences (a) Bengali word-level parse tree (b) Hindi word-level parse tree

Figure 4: Chunk-level parse trees of the example Bengali and Hindi sentences (a) Bengali chunk-level parse tree (b) Hindi chunk-level parse tree

Step 4: This parser was transferred using the methods described in section 5.2 to get the delexicalized parser for Bengali head trees.

Figure 5: Chunk-level parse tree of the example Bengali sentence before and after expansion (a) Bengali chunk head parse tree (b) Bengali chunk head parse tree after expansion

Step 5: For testing, we parsed the Bengali test sentences consisting of only the chunk head words. The UAS score for head trees obtained by delexicalized method is 68.6.

Step 6: For intra-chunk expansion we simply attached the non-head words to their corresponding chunk heads to get the full trees (This introduces a lot of errors. In future we plan to use rules for chunk expansion to make the intra-chunk expansion more accurate.) The UAS score for trees after intra-chunk expansion is 78.2.

We observed that our simple heuristic for inter-chunk expansion increases accuracy of the parser. There are some rule-based methods and statistical approach for inter-chunk expansion (Kosaraju et al., 2012; Bharati et al., 2009a; Chatterji et al., 2012) in Hindi which may be adopted for Bengali.

Table 3: Comparison of word-level and chunk-level transfer of parse trees

	Delexicalized	Projected Bengali vectors
Trees after word-level transfer	65.1	67.2
Expanded chunk head trees after chunk-level transfer	78.2	75.8

Table 3 compares the UAS of baseline parsers for word-level transfer with chunk-level transfer followed by expansion. We found a significant increase of UAS score from 65.1 to 78.2 after parsing and subsequent intra-chunk expansion. However, while using common vector-based word representation had shown slight improvement when applied to the word level transfer it did not help when applied to chunk level transfer. This may be because we used only the vector embeddings of chunk heads for the chunk-level parsing. We wish to work further on vector representation of chunks which might capture more chunk-level information and help improve the results.

While chunking has been used with other parsers, we did not find any work that uses chunking in a transfer parser. The source (Hindi) delexicalized word-level parser gave an accuracy of 77.7% and the source (Hindi) delexicalized chunk-level parser followed by expansion gave an accuracy of 79.1% on the UD Hindi test data.

There is no reported work on cross-lingual transfer between Bengali and Hindi. But as a reference we will like to mention the type of UAS accuracy values reported for other transfer parsers based on delexicalization in the literature for other language pairs. Zeman and Resnik (2008)'s delexicalized parser gave a F-score of 66.4 on Danish language. Täckström et al. (2012) achieved an average UAS of 63.0 by using word clusters on ten target languages and English as the source language. They achieved UAS of 57.1 without using any word cluster feature. In their works, (Xiao and Guo, 2014) tried out cross-lingual parsing on a set of eight target languages with English as the source language and achieved a UAS of 58.9 on average while their baseline delexicalized MSTParser parser using universal POS tag features gave an UAS of 55.14 on average. Duong et al. (2015b) also applied their method on nine target languages and English as the source language. They achieved UAS of 58.8 on average.

6 Error analysis

We analyzed the errors in dependency relations of the parse trees obtained by parsing the test sentences. We analyze the results based on the number of dependency relations in the gold data that actually appear in the trees parsed by our parser. We report results of the ten most frequent dependency tags in table 4.

From table 4 we find that chunk-level transfer increases the accuracy of tree root identification. Chunk-level transfer significantly increases the ac-

Table 4: Comparison of errors for 12 dependency tags. The entries of column 3 to 6 indicates the number of dependencies bearing the corresponding tags in the gold data that actually appear in the parsed trees and the accuracy (in %).

	Actual Count of dependency relations	Delexicalized word-level transfer		Word-level transfer using projected Bengali vectors		Delexicalized chunk-level transfer		Chunk-level transfer using projected Bengali vectors	
k1 (doer/agent/subject)	166	111	(66.9)	104	(62.7)	133	(80.1)	118	(71.1)
vmod (Verb modifier)	111	71	(64.0)	78	(70.3)	85	(76.6)	71	(64.0)
main (root)	150	96	(64.4)	108	(72.5)	105	(70.5)	103	(69.1)
k2 (object)	131	100	(76.3)	92	(70.2)	104	(79.4)	88	(67.2)
r6 (possessive)	82	21	(25.6)	49	(59.8)	13	(15.9)	52	(63.4)
pof (Part of relation)	59	55	(93.2)	58	(98.3)	56	(94.9)	56	(94.9)
k7p (Location in space)	50	31	(62.0)	30	(60.0)	38	(76.0)	33	(66.0)
ccof (co-ordinate conjunction of)	47	1	(2.1)	4	(8.5)	1	(2.12)	2	(4.3)
k7t (Location in time)	40	25	(62.5)	20	(50.0)	31	(77.5)	15	(37.5)
k7 (Location elsewhere)	22	15	(68.2)	14	(63.6)	16	(72.7)	17	(77.3)
k1s (noun complement)	18	13	(72.2)	14	(77.8)	14	(77.8)	14	(77.8)
relc (relative clause)	12	1	(8.4)	1	(8.4)	0	(0.0)	0	(0.0)

curacy of identifying the relations with *k1*, *vmod*, *k2* and *k7* tags also.

Although delexicalized chunk-level parser gives the overall best result, the accuracy is lowest for the relation of type *r6* (possessive/genitive). We observed that in most of the erroneous cases, both the words that are expected to be connected by the *r6* dependency, are actually being predicted as modifiers of a common parent. We find that the accuracy of *r6* tag improves in case of delexicalized word-level transfer and the best accuracy on *r6* is achieved with the use of lexical features. Hence, the drop in performance may be due to the lack of sufficient information in the case of chunk-level transfer or the chunk expansion heuristic that we have used this work.

However, for all the methods discussed above the parser performs poorly in identifying the "conjuction of" (*ccof*) relations and relative clause (*relc*) relations. we observed that the poor result on *ccof* tag is due to the difference in annotation scheme of ICON and UD. In case of ICON data, the conjunctions are the roots of the trees and the corresponding verbs or nouns are the modifiers, while in UD scheme the conjunctions are the modifiers of the corresponding verbs of nouns. We need to investigate further into the poor identification of *relc* dependencies.

7 Conclusion

We show that knowledge of shallow syntactic structures of the languages helps in improving the quality of cross-lingual parsers. We observe that chunking significantly improves cross-lingual parsing from Hindi to Bengali due to their syntactic similarity at the phrase level. The experimental results clearly shows that chunk-level transfer of parser model from Hindi to Bengali is better than direct word-level transfer. This also goes to establish that one can improve the performance of pure statistical systems if one additionally uses some linguistic knowledge and tools. The initial experiments were done in Bengali. In future we plan to broaden the results to include other Indian languages for which open source chunkers can be found.

References

Bharat Ram Ambati, Samar Husain, Sambhav Jain, Dipti Misra Sharma, and Rajeev Sangal. 2010. Two methods to incorporate 'local morphosyntactic' features in Hindi dependency parsing. In *Proceedings of the NAACL HLT 2010 First Workshop on SPMRL*, pages 22–30, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Bharat Ram Ambati, Rahul Agarwal, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. 2011. Error detection for treebank validation. *Asian Language Resources collocated with IJCNLP 2011*, page 23.

Rafiya Begum, Karan Jindal, Ashish Jain, Samar Husain, and Dipti Misra Sharma. 2011. Identification of conjunct verbs in hindi and its effect on parsing accuracy. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 29–40. Springer.

Akshar Bharati and Rajeev Sangal. 1993. Parsing free word order languages in the paninian framework. In *Proceedings of the 31st Annual Meeting on ACL*, ACL '93, pages 105–111, Stroudsburg, PA, USA. Association for Computational Linguistics.

Akshar Bharati, Rajeev Sangal, and T Papi Reddy. 2002. A constraint based parser using integer programming. *Proc. of ICON*.

Akshar Bharati, Mridul Gupta, Vineet Yadav, Karthik Gali, and Dipti Misra Sharma. 2009a. Simple parser for indian languages in a dependency framework. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 162–165, Suntec, Singapore, August. Association for Computational Linguistics.

Akshar Bharati, Samar Husain, Meher Vijay, Kalyan Deepak, Dipti Misra Sharma, and Rajeev Sangal. 2009b. Constraint based hybrid approach to parsing indian languages. In *Proceedings of the 23rd PACLIC*, pages 614–621, Hong Kong, December. City University of Hong Kong.

Rajesh Bhatt and Fei Xia. 2012. Challenges in converting between treebanks: a case study from the hutb. In *META-RESEARCH Workshop on Advanced Treebanking*, page 53.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 186–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Samit Bhattacharya, Monojit Choudhury, Sudeshna Sarkar, and Anupam Basu. 2005. Inflectional morphology synthesis for bengali noun, pronoun and verb systems. In *In Proceedings of the national conference on computer processing of Bangla (NCCPB*, pages 34–43.

Sanjay Chatterji, Praveen Sonare, Sudeshna Sarkar, and Devshri Roy. 2009. Grammar driven rules for hybrid bengali dependency parsing. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing, Hyderabad, India*.

Sanjay Chatterji, Arnad Dhar, Sudeshna Sarkar, and Anupam Basu. 2012. A three stage hybrid parser for hindi. In *Proceedings of the Workshop on MTPIL*, pages 155–162, Mumbai, India, December. The COLING 2012 Organizing Committee.

Sanjay Chatterji, Tanaya Mukherjee Sarkar, Pragati Dhang, Samhita Deb, Sudeshna Sarkar, Jayshree Chakraborty, and Anupam Basu. 2014. A dependency annotation scheme for bangla treebank. *Language Resources and Evaluation*, 48(3):443–477.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 EMNLP*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

S. Dandapat, S. Sarkar, and A. Basu. 2004. A hybrid model for part-of-speech tagging and its application to bengali.

Arjun Das, Arabinda Shee, and Utpal Garain. 2012. Evaluation of two bengali dependency parsers. In *Proceedings of the Workshop on MTPIL*, pages 133–142, Mumbai, India, December. The COLING 2012 Organizing Committee.

Sankar De, Arnab Dhar, and Utpal Garain. 2009. Structure simplification and demand satisfaction approach to dependency parsing for bangla. In *Proc. of 6th ICON tool contest: Indian Language Dependency Parsing*, pages 25–31.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015a. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *Proceedings of the Nineteenth Conference on CONLL*, pages 113–122, Beijing, China, July. Association for Computational Linguistics.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015b. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. *Volume 2: Short Papers*, page 845.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on EMNLP and CoNLL*, pages 1–11, Jeju Island, Korea, July. Association for Computational Linguistics.

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. ACL '09, pages 369–377, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aniruddha Ghosh, Pinaki Bhaskar, Amitava Das, and Sivaji Bandyopadhyay. 2009. Dependency parser for bengali: the ju system at icon 2009. In *Proc. of 6th ICON tool contest: Indian Language Dependency Parsing*, pages 7–11.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd ACL and the 7th IJCNLP*, volume 1, pages 1234–1244.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:11–311.

Prudhvi Kosaraju, Sruthilaya Reddy Kesidi, Vinay Bhargav Reddy Ainavolu, and Puneeth Kukkadapu. 2010. Experiments on indian language dependency parsing. *Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing*.

Prudhvi Kosaraju, Bharat Ram Ambati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal. 2012. Intra-chunk dependency annotation : Expanding

hindi inter-chunk annotated treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 49–56, Jeju, Republic of Korea, July. Association for Computational Linguistics.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 1337–1348, Baltimore, Maryland, June. Association for Computational Linguistics.

Prashanth Mannem. 2009. Bidirectional dependency parser for hindi, telugu and bangla. *Proceedings of ICON09 NLP Tools Contest: Indian Language Dependency Parsing, Hyderabad, India*.

Ryan Mcdonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *In Proceedings of HLT Conference and Conference on EMNLP*, pages 523–530.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on EMNLP*, EMNLP '11, pages 62–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS 26*, pages 3111–3119. Curran Associates, Inc.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the ACL: Long Papers - Volume 1*, ACL '12, pages 629–637, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, may.

Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical report, Växjö University.

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

Li-Shiuan Peh and Christopher Ting Hian Ann. 1996. A divide-and-conquer strategy for parsing. *CoRR*, cmp-lg/9607020.

Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the 2015 Conference on EMNLP*, pages 328–338, Lisbon, Portugal, September. Association for Computational Linguistics.

Agnivo Saha and Sudeshna Sarkar. 2016. Enhancing neural network based dependency parsing using morphological information for hindi. In *17th CICLing*, Konya, Turkey, April. Springer.

D.M. Sharma, Sangal R., L. Bai, R. Begam, and K. Ramakrishnamacharyulu. 2007. Anncorra : Treebanks for indian languages, annotation guidelines (manuscript).

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the NAACL: HLT*, NAACL HLT '12, pages 477–487, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oscar Täckström, Ryan T. McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. pages 1061–1071.

Jörg Tiedemann. 2015. Improving the cross-lingual projection of syntactic dependencies. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 191–199, Vilnius, Lithuania, May. Linköping University Electronic Press, Sweden.

Ashwini Vaidya, Owen Rambow, and Martha Palmer. 2014. Light verb constructions with 'do'and 'be'in hindi: A tag analysis. In *Workshop on Lexical and Grammatical Resources for Language Processing*, page 127.

Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*.

Min Xiao and Yuhong Guo. 2015. Annotation projection-based representation learning for cross-lingual dependency parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 73–82, Beijing, China, July. Association for Computational Linguistics.

D. Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. *NLP for Less Privileged Languages*, pages 35 – 35.