A Statistical Approach to Prediction of Empty Categories in Hindi Dependency Treebank

Puneeth Kukkadapu, Prashanth Mannem
Language Technologies Research Center
IIIT Hyderabad, India
{puneeth.kukkadapu,prashanth}@research.iiit.ac.in

Abstract

In this paper we use statistical dependency parsing techniques to detect NULL or Empty categories in the Hindi sentences. We have currently worked on Hindi dependency treebank which is released as part of COLING-MTPIL 2012 Workshop. Earlier Rule based approaches are employed to detect Empty heads for Hindi language but statistical learning for automatic prediction is not explored. In this approach we used a technique of introducing complex labels into the data to predict Empty categories in sentences. We have also discussed about shortcomings and difficulties in this approach and evaluated the performance of this approach on different Empty categories.

1 Introduction

Hindi is a morphologically rich and a relatively free word order language (MoR-FWO). Parsing is a challenging task for such MoR-FWO languages like Turkish, Basque, Czech, Arabic, etc. because of their non-configurable nature. Previous research showed that the dependency based annotation scheme performs better than phrase based annotation scheme for such languages (Hudson, 1984; Bharati et al., 1995). Dependency annotation for Hindi is based on Paninian framework for building the treebank (Begum et al., 2008). In recent years data driven parsing on Hindi has shown good results, the availability of annotated corpora is a definite factor for this improvement (Nivre et al., 2006; McDonald et al., 2005; Martins et al., 2009; Mannem and Dara, 2011). Other approaches such as rule-based and hybrid of rule-based and data-driven (Bharati et al., 2009a) for Hindi language have also been tried out. In the shared task for Hindi Parsing organized with COLING workshop Singla et al. (2012) achieved best results for Gold-Standard data with 90.99% (Labeled Attachment Score or LAS) and 95.87% (Unlabeled Attachment Score or UAS).

Empty category is a nominal element which does not have any phonological content and is therefore unpronounced. Empty categories are annotated in sentences to ensure a linguistically plausible structure. Empty categories play a crucial role in the annotation framework of the Hindi dependency treebank (Begum et al., 2008; Bharati et al., 2009b). If dependency structure of a sentence do not form a fully connected tree then Empty category (denoted by NULL in Hindi Treebank) is inserted in the sentence. In the Hindi treebank, an Empty category has at least one child. Traditional parsing algorithms do not insert Empty categories and require the Empty categories to be part of the input. These Empty categories are manually annotated in the treebank. In real time scenarios, like translation between languages, it is not possible to add the Empty categories into the sentences manually. So we require an approach which can identify the presence of these Empty categories and insert into appropriate positions in the sentence.

Figure 1 shows an Example of a Hindi sentence annotated with a NULL category. The English translation for this sentence is, “Its not fixed what his big bank will do”. The aim of this paper is to investigate the problem of automatically predicting the Empty categories in the sentences using the statistical de-

Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages, pages 91–96,
Seattle, Washington, USA, 18 October 2013. ©2013 Association for Computational Linguistics

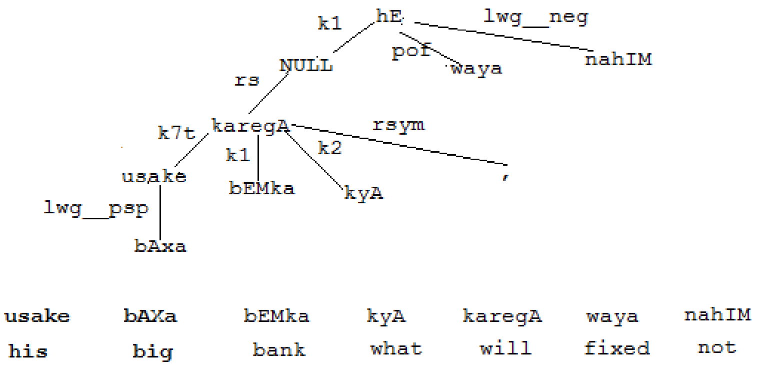

Figure 1: An Example of a Hindi sentence annotated with a NULL category.

pendency parsing technique and to shed some light on the challenges of this problem. As the data-driven parsing on Hindi language has achieved good results (Singla et al., 2012), we try to use this approach to predict Empty categories in the sentence. In this approach the information about NULL categories is encoded into the label set of the structure. In these experiments we have used only Projective sentences from the treebank. Non-projectivity makes it difficult to identify the exact position of NULLs during introduction of NULLs in the sentence.

The rest of the paper is divided into the following sections: Section 2 discusses about the related work. Section 3 gives an overview of the Hindi data we have used for our experiments. Section 4 contains the details of our approach and section 5 discusses about experiments, parser, results and discussion. We conclude the paper in section 6 with a summary and the future work.

2 Related Work

Previous work related to Empty categories prediction on Hindi data is done by Gsk et al. (2011) which is a rule based approach for detection of Empty categories and also presented detailed analysis of different types of Empty categories present in the Hindi treebank. They used hand-crafted rules in order to identify each type of Empty category. As this is a rule based approach it becomes language specific. There are many approaches for the recovery of empty categories in the treebanks like Penn treebank, both ML based (Collins, 1997; Johnson, 2002; Seeker et al., 2012), and rule based (Campbell, 2004). Some approaches such as Yang and Xue (2010) follow a post processing step of recovering empty categories after parsing the text. Gsk et al. (2011) have discussed about different types of Empty categories in Hindi Treebank in detailed manner. The main types of Empty categories are:

- Empty Subject where a clause is dependent on missing subject (NP) of the verb, denoted as NULL_NP or NULL_PRP.
- Backward Gapping where the verb (VM) is absent in the clause that occurs before a coordinating conjunct, denoted as NULL_VM
- Forward Gapping where the verb (VM) is absent in the clause that occurs after a coordinating conjunct, denoted as NULL_VM.
- Conjunction Ellipses where the Conjunction (CC) is absent in the sentence, denoted as NULL_CC.

3 Data

We have used COLING-MTPIL workshop 2012 data for our experiments. This was released by the organizers as part of the shared task in two different settings. One being the manually annotated data with POS tags, chunks and other information such as gender, number, person etc. whereas the other one contains only automatic POS tags without any other information. We have used Gold standard data with

Type of NULL	No. of Instances
NULL_VM	247
NULL_CC	184
NULL_NP	71
NULL_PRP	25

Table 1: Empty categories in Training + Development Dataset of Hindi treebank.

Type of NULL	No. of instances
NULL_VM	26
NULL_CC	36
NULL_NP	9
NULL_PRP	4

Table 2: Empty categories in Testing Dataset of Hindi treebank.

all features provided for our experiments. Training set contains 12,041 sentences, development data set consists of 1233 sentences and testing data set consists of 1828 sentences. In our experiments we have worked with only projective sentences. We have combined the training and development data sets into one data set and used as training in the final experiments.

Training and Development data together consists of 544 NULL instances (in 436 sentences) of 10,690 sentences. The major types of Empty categories present in the training data are of type NULL_CC, NULL_VM, NULL_NN and NULL_PRP categories. Table 1 and Table 2 show the number of instances of each category. Testing data consists of 80 instances (72 sentences) of 1455 sentences.

4 Approach

There are 3 main steps involved in this process.

4.1 Pre-Processing

In the first step, we encode information about presence of Empty categories in a sentence into the dependency relation label set of the sentence. If NULLs are present in a sentence, we remove the NULLs from the respective sentence in the treebank. In a sentence the dependents or children of a NULL category are attached to the parent of the NULL category and their respective labels are combined with dependency label of NULL category which indicates the presence of NULL and also says that such words or tokens are children of NULL category. Instead of just combining the labels we also add a sense of direction to the complex label which indicates whether the position of NULL is to the right or left of this token in the sentence and subsequently NULLs are also detached from its parent node. Therefore a complex label in a sentence indicates the presence of a NULL category in the sentence.

Example: **Null-label_r_dep-label** is a generic type of a complex label. In this format 'r' indicates that a NULL instance is to the right of this token. Null-label is the dependency relation label joining the Null instance and its parent and dep-label is the dependency relation label joining the current token or word to its parent which is a NULL instance. Figure 2 illustrates this step.

4.2 Data-driven parsing

In the second step a Data-driven parser is trained using the training data (with complex dependency relation labels) and when this parser model is used on the test data it predicts the complex labels in the output. In this approach we have tried out different data-driven parsers such as Malt (Nivre et al., 2006), Turbo (Martins et al., 2010) and MST (McDonald et al., 2005) for this experiment which were shown earlier to be performing better for Hindi Parsing by Kukkadapu et al. (2012) and found that Malt parser performs better than the rest on this data with complex labels.

4.3 Post-processing

In the final step, Post-processing is applied on the output predicted by the parser in the above step. In this step presence of NULLs are identified using the complex labels and their position in the sentence is identified using sense of direction in these labels (i.e., whether NULL instance is to the left 'l' or right 'r' of this token). During the insertion of NULLs into the sentence Projectivity of the sentence must be preserved. Keeping this constraint intact and using the direction information from the dependency relation labels, NULLs are introduced into the sentence. Figure 2 illustrates this step.

The advantage in using statistical approach rather than a rule based approach to predict NULLs is, it

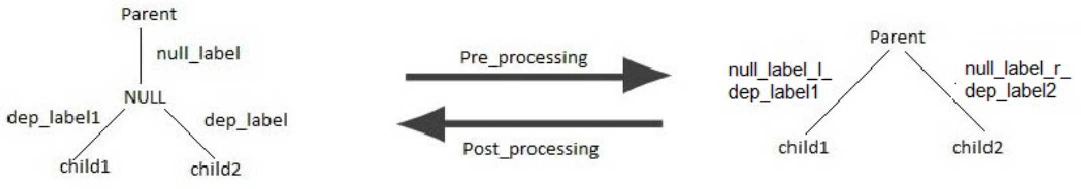

Figure 2: Process

can be easily used to predict NULLs in other MoR-FWO languages. The problem with this approach is, it can't handle Empty categories occurring as Leaf nodes (or Terminal nodes in the dependency tree) and as Root nodes. As we have mentioned earlier, the dependency annotation scheme of Hindi language does not allow for Empty categories to occur as Leaf nodes (or Terminal nodes). But if these Empty categories occur as Root nodes in the dependency tree then such cases are not disturbed in our approach.

5 Experiments and Results

5.1 Parser settings

As mentioned earlier we had used Malt parser for our experiments. Malt Parser implements the transition based approach to dependency parsing which has two components:
1) A transition system for mapping sentences into dependency trees.
2) A classifier for predicting the next transition for every possible system configuration.

Malt parser provides two learning algorithms LIBSVM and LIBLINEAR. It also provides various options for parsing algorithms and we have experimented on nivre-eager, nivre-standard and stack-proj parsing algorithms. Nivre-eager has shown good results in our experiments.

5.2 Features and Template

Feature model is the template, which governs the learning from the given training data. We observed feature model used by Kosaraju et al. (2010) performs best.

In order to get best results in the second step (Data-driven parsing) we have experimented with various features provided in the data. Kosaraju et al. (2010) and Husain et al. (2010) showed the best features that can be used in FEATS column in CoNLL-X format. These features are vibhakti (post positional marker), TAM (tense, aspect and modality), chunk features like chunk head, chunk distance and chunk boundary information have proved to be effective in parsing of Hindi language and our results on overall accuracy of data is consistent with their results.

Type of NULL Category	Recall
NULL_VM	50
NULL_CC	69.45
NULL_NN	88.89
NULL_PRP	50

Table 3: Empty categories Predicted by this approach on test data.

5.3 Results and Discussion

The Results obtained on the test dataset are shown below and Recall on each Empty category are given in Table 3:

The Results obtained by using this approach on the test set including all the Empty category types is as follows:

Precision = **84.9**
Recall = **69.23**
F-measure = **76.26**

In computation of the above results the exact position of NULLs in the sentence are not considered. These values indicate the efficiency of the system in identifying the presence of the Empty categories in the system. However, this approach inserted the

NULLs in exact positions with a Precision of more than 85%, i.e., of all the NULL instances it has inserted correctly, it has inserted 85% of them in exact positions in the sentences.

The approach was able to insert NULL_NP tokens with good accuracy but it had a tough time predicting NULL_VM tokens. This was also consistent with Gsk et al. (2011) conclusions about Empty categories in Hindi treebank.

In case of NULL_VM categories we have observed some inconsistency in the annotation of these sentences. In these sentences which have multiple clauses with main verb (VM) token missing, certain sentences are annotated with NULL_VM for each clause where main verb (VM) token is missing and certain sentences are annotated with one NULL_VM for all the clauses with main verb (VM) missing. This may be a reason for accuracy drop in predicting NULL_VM tokens. The main reason for low accuracy as we have observed is that the output predicted by the parser is low for these complex labels. The test data consists of 202 complex labels whereas the parser has been able to predict only 102 of them, which is a huge drop in accuracy for complex labels. The overall accuracy of parser on the test data (only projective sentences) has been high 91.11%(LAS), 95.86%(UAS) and 92.65%(LS). The low accuracy of the parser on complex labels may be due to less number of these instances compared to size of the corpus. Another reason may be due to the introduction of complex labels the size of label set has increased significantly and it may be difficult for the parser to learn the rare labels.

6 Conclusion and Future work

In this paper, we presented a statistical approach to Empty category prediction using Data-driven parsing. We have used state-of-the-art parser for Hindi language with an accuracy above 90% and have achieved a decent F-score of 76.26 in predicting Empty categories. We look to try out this approach for other MoR-FWO languages and compare the performances on different languages. We need to identify Features which would help in identifying NULL_CC category and also should try this approach on a big data set with a significant number of instances of NULLs and also look to extend this approach to Non-Projective sentences.

References

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*.

A. Bharati, V. Chaitanya, R. Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: A Paninian perspective*. Prentice-Hall of India.

Akshar Bharati, Samar Husain, Dipti Misra, and Rajeev Sangal. 2009a. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 77–80. Association for Computational Linguistics.

Akshara Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begam, and Rajeev Sangal. 2009b. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank.

Richard Campbell. 2004. Using linguistic principles to recover empty categories. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 645. Association for Computational Linguistics.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 16–23. Association for Computational Linguistics.

Chaitanya Gsk, Samar Husain, and Prashanth Mannem. 2011. Empty categories in hindi dependency treebank: Analysis and recovery. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 134–142. Association for Computational Linguistics.

R.A. Hudson. 1984. *Word grammar*. Blackwell Oxford.

Samar Husain, Prashanth Mannem, Bharat Ram Ambati, and Phani Gadde. 2010. The icon-2010 tools contest on indian language dependency parsing. *Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing, ICON*, 10:1–8.

Mark Johnson. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 136–143. Association for Computational Linguistics.

P. Kosaraju, S.R. Kesidi, V.B.R. Ainavolu, and P. Kukkadapu. 2010. Experiments on indian language dependency parsing. *Proceedings of the ICON10 NLP Tools Contest: Indian Language Dependency Parsing*.

Puneeth Kukkadapu, Deepak Kumar Malladi, and Aswarth Dara. 2012. Ensembling various dependency parsers: Adopting turbo parser for indian languages. In *24th International Conference on Computational Linguistics*, page 179.

P. Mannem and A. Dara. 2011. Partial parsing from bitext projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1597–1606.

A.F.T. Martins, N.A. Smith, and E.P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 342–350.

A.F.T. Martins, N.A. Smith, E.P. Xing, P.M.Q. Aguiar, and M.A.T. Figueiredo. 2010. Turbo parsers: Dependency parsing by approximate variational inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 34–44.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–530.

J. Nivre, J. Hall, and J. Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

Wolfgang Seeker, Richárd Farkas, Bernd Bohnet, Helmut Schmid, and Jonas Kuhn. 2012. Data-driven dependency parsing with empty heads. In *Proceedings of COLING 2012: Posters*, pages 1081–1090, Mumbai, India, December. The COLING 2012 Organizing Committee.

Karan Singla, Aniruddha Tammewar, Naman Jain, and Sambhav Jain. 2012. Two-stage approach for hindi dependency parsing using maltparser. *Training*, 12041(268,093):22–27.

Yaqin Yang and Nianwen Xue. 2010. Chasing the ghost: recovering empty categories in the chinese treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1382–1390. Association for Computational Linguistics.