Using Lexical and Dependency Features to Disambiguate Discourse Connectives in Hindi

Rohit Jain, Himanshu Sharma, Dipti Misra Sharma
FC Kohli Center on Intelligent Systems (KCIS)
International Institute of Information Technology - Hyderabad
Telangana, India
rohit.jain@research.iiit.ac.in, himanshu.sharma@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

Discourse parsing is a challenging task in NLP and plays a crucial role in discourse analysis. To enable discourse analysis for Hindi, Hindi Discourse Relations Bank was created on a subset of Hindi TreeBank. The benefits of a discourse analyzer in automated discourse analysis, question summarization and question answering domains has motivated us to begin work on a discourse analyzer for Hindi. In this paper, we focus on discourse connective identification for Hindi. We explore various available syntactic features for this task. We also explore the use of dependency tree parses present in the Hindi TreeBank and study the impact of the same on the performance of the system. We report that the novel dependency features introduced have a higher impact on precision, in comparison to the syntactic features previously used for this task. In addition, we report a high accuracy of 96% for this task.

Keywords: Discourse analysis, Discourse connective identification, Hindi

1. Introduction

Units within a piece of text are not meant to be understood independently but understood by linking them with other units in the text. These units may be clauses, sentences or even complete paragraphs. Establishing relations between units present in a text allows the text to be semantically well structured and understandable. Understanding the internal structure of text and the identification of discourse relations is called discourse analysis.

The Hindi Discourse Relation Bank (Prasad et al., 2008b; Oza et al., 2009; Kolachina et al., 2012) was created to enable discourse analysis and research beyond sentence-level for Hindi. Similar discourse annotation projects for English (Miltsakaki et al., 2004; Prasad et al., 2008a), Chinese (Xue, 2005; Zhou and Xue, 2012), Czech (Mladová et al., 2008; Poláková et al., 2013) and Turkish (Zeyrek and Webber, 2008; Zeyrek et al., 2010) have also been carried out.

A fully automated discourse analyzer would thus greatly aid discourse analysis and NLP applications such as text summarization and question answering. The benefits of a fully automated discourse analyzer has led us to work towards the same for Hindi.

Given a text, a discourse analyzer would mark the presence discourse relations between two spans of text in a discourse. The presence of discourse relations are often marked by the presence of discourse connectives. Such discourse relations are called Explicit discourse relations, and those whose presence is not marked by discourse connectives are called Implicit discourse relations.

Thus the first component towards a complete discourse analyzer would be discourse connective identifier. Our work involves identification of discourse connectives which are explicitly realized i.e explicit connectives.

In this paper we experiment with lexical and syntactic features and study their performance for this task. In addition to these features we attempt to make use of the dependency tree parses in the Hindi TreeBank (Begum et al., 2008) and interpolated models to further improve performance. We report a high accuracy of 96% for this task and also report significant improvements in performance because of novel dependency features.

The rest of the paper is organized as follows. Section 2. briefly introduces HDRB and describes the task. We discuss related work and our approach in Section 3. Experiments and results are presented in Section 4. Finally, we conclude in Section 5.

2. Corpus and Task description

2.1. Hindi Discourse Relation Bank

The Hindi Discourse Relation Bank(HDRB) was previously created broadly following the lines of Penn Discourse TreeBank (PDTB)(Miltsakaki et al., 2004; Prasad et al., 2008a)'s lexically grounded approach along with a modified annotation workflow, additional grammatical categories for explicit connectives, semantically driven Arg1/Arg2 labelling and modified sense hierarchies.(Oza et al., 2009; Kolachina et al., 2012)

HDRB was annotated on a subset of the Hindi TreeBank (Begum et al., 2008) which includes part-of-speech, chunk and dependency parse tree annotations. In comparison, Penn TreeBank (Marcus et al., 1993) has parts-of-speech, chunk and constituent parse tree annotations.

The dependency annotations scheme in Hindi treats a sentence as a series of modifier-modified relations. Thus the root of the dependency tree would be the primary modified of the sentence which is generally the main verb of the sentence. The participant relations with the verb are called karakas. There are six basic karaka relations namely *adhikarana*(location), *apaadaan*(source), *sampradaan*(recipient), *karana*(instrument), *karma*(theme) and *karta*(agent).

HDRB contains 1865 sentences and a word count of 42K. Furthermore HDRB contains 650 explicit discourse relations and 1200 implicit discourse relations. Out of 125 connectives annotated in HDRB, 22 occur as discourse connectives more than 90% of the time.

2.2. Task description

Hindi Discourse Relation Bank(HDRB) contains a total of 125 annotated discourse connectives. Many of these 125 discourse connectives occur in non-discourse context in the corpus. For example:

(1) लघु उद्योग के लिए नई योजना नहीं है **और** कस्टम ड्यूती में कटौती का सीधा असर घरेलू उद्योग पर पडेगा।

There is no new project for small business **and** cut-offs in the custom duty would have a direct effect on the domestic businesses.

(2) जॉन **और** मैरी बाज़ार गए थे।

John **and** Mary went to the market.

In sentence (1) *and* is a discourse connective between two clauses whereas in sentence (2) *and* occurs in a non-discourse context. Our task is thus to differentiate between the discourse and non-discourse usage of a given connective.

3. Approach

Existing work on discourse connective identifiers for English (Pitler and Nenkova, 2009; Lin et al., 2014; Wang and Lan, 2015) employ a classifier based approach with differences arising in choice of feature sets. We have decided to adopt the same classifier based approach for our task.

We thus approach the task as classifying a set of connectives as positive or negative, where positive connectives are connectives marking the presence of a discourse relation and negative connectives are connectives marking no such presence. The set of possible connectives is obtained using a list of 125 known discourse connectives.

Selection of features is a crucial aspect of classifier based approaches. Since we are to identify positive connectives from free text, our aim while selecting features should be to capture information regarding the connective and its neighboring words. We should also aim to capture the position and role of connective in the dependency tree to further improve performance. We first discuss the syntactic features we have considered. Later on, we discuss our approaches and features to capture information regarding the position and role of the connective in the dependency tree.

C-String The connective itself would be an important feature to differentiate between positive and negative connectives. Since the possible set of connectives has been generated using a fixed set of known discourse connectives, it might seem redundant to include connective string as a feature. However including the connective string would allow the model to better understand the distribution of features with respect to each connective.

C-Pos The connective's POS tag will be an indicator of the basic grammatical role of the connective.

C-Chunk The tag of the chunk containing the connective will be an indicator of the larger role played by the connective in the sentence.

C-POS-Neighbour [+/-][1,2] The POS tag of previous word[-] and word occurring after the connective[+] and the distance[1,2] of the same from the connective. These features form the basic description of the syntactic neighborhood of the connective.

C-Chunk-Neighbour [+/-][1,2] The chunk tag of previous chunk[-] and chunk occurring after the connective[+] and the distance[1,2] of the same from the connective. The motivation behind using these features is the same as discussed before i.e. to capture information regarding syntactic neighborhood and larger role played in the sentence.

Apart from using features similar to the above mentioned features, Lin et al. (2014) also used two additional features namely "Path of connective parent to root" and "Compressed path of connectives parent to root" in order to capture the syntactic relation between the connective and the syntactic root.

HDRB has dependency tree annotations and thus we attempt to capture the semantic relation between connective and root of dependency tree through similar features. However due to a smaller corpus, we believe such simple features will not necessarily capture the information provided by the dependency annotations. Thus to capture semantic information, we introduce *Dependency Relation* feature. Furthermore, to extract information regarding the position of the connective in the dependency tree we introduce 3 novel features namely *Coordination between clauses*, *Right Argument Location* and *Left Argument Location*.

Dependency Relation Dependency relation of previous chunk, connective chunk and chunk after connective chunk with their respective parents in the dependency tree. Thus rather than capture the semantic relation between chunk in question and root of dependency tree, we limit it to the immediate parent, which in most cases is the verb of the clause the chunk is present in. We attempt to capture the semantic role played by the connective chunk and its neighboring chunks through this feature.

Coordination between clauses This features checks whether the connective has two clauses as its children in the dependency tree. We attempt to capture information regarding whether the connective in question plays the role of a coordinator between two clauses.

This features is difficult to implement without the dependency tree since it is not a trivial task to identify the verb clauses which are coordinated by the connective. The presence of verb clauses before and after the connective in the sentence does not necessarily indicate the kind of coordination we are attempting to capture. A simple analysis of connective behaviour shows that coordinating conjunctions(and, but) display such behaviour and account for 30% of the connectives.

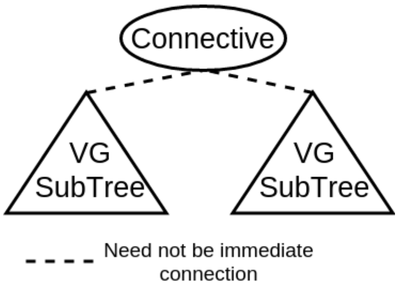

Figure 1: Coordination between clauses

Right Argument Location A discourse relation consists of a discourse connective and two arguments namely Arg1 and Arg2. Arg2 occurs immediately after the connective. This feature labels the location of word immediately after the connective with respect to position of the connective in the dependency tree. We attempt to capture the possibility of Arg2 being present in that position if the connective was playing the role of a discourse connective. For the remainder of this description we refer to the first word of Arg2 i.e. the word immediately after the connective as rightArg. The various position of rightArg as follows:

Last connective occurs at the end of the sentence i.e. no rightArg

Same Node rightArg occurs in the same node as connective

Direct Parent rightArg and connective share as immediate parent

Conn Parent rightArg is present in the connective's sub-tree

rightArg Parent connective is present in the rightArg's sub-tree

Indirect Conn Parent connective's parent node sub tree contains rightArg

Indirect rightArg Parent rightArg's parent node sub-tree contains connective

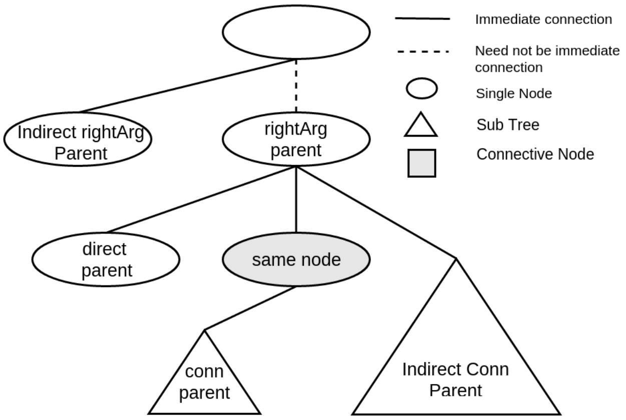

Figure 2: rightArg Locations

Left Argument Location A discourse relation consists of a discourse connective and two arguments namely Arg1 and Arg1. This feature, similar to right argument location feature, labels the position of the word occurring just before the connective. Unlike Arg2, which is always occurs after the connective, there is no such constraint for Arg1. However a simple analysis shows that Arg1 occurs immediately before the connective in 45% of the cases. We attempt to capture the possibility of Arg1 being present in that position if the connective was playing the role of a discourse connective. We refer to the last word of Arg1 i.e. the word immediately before the connective to be leftArg. The various positions of leftArg are the same as that of rightArg.

4. Experiments and Results

We conduct several experiments to arrive at the best feature set for our task. We also experimented with Maximum Entropy (Fan et al., 2008), Naive-Bayes (Zhang, 2004) and Conditional Random Fields (Lafferty et al., 2001) to train the classifier.

We now report the performance of our chosen feature set using Maximum Entropy using ten-fold cross validation. To put this comparison into better perspective, we implement the feature sets chosen by (Pitler and Nenkova, 2009) and (Lin et al., 2014) for Hindi and then compare the performance of the respective feature sets to our feature set.

4.1. Performance of Chosen Feature Set

The performance of our feature set is shown in Table 1.

The baseline using only C-String feature results in a reasonably high f-measure of 73.05%. Using C-Chunk, C-ChunkN[-] and C-ChunkN[+] features resulted in significant improvements over the baseline(73.05% to 88.6%). Similar performance improvements over the baseline is achieved while using C-POS, C-POSN[-] and C-POSN[+] as features(73.05% to 87.79%). Using both sets of features further improves performance over baseline(73.05% to 90.05%).

Thus using purely lexical features we have achieved an f-measure of 90%. We now study the impact of the dependency features and henceforth refer to the performance achieved by lexical features as LEX_BASE i.e. lexical baseline.

Using each of *Dependency Relation*, *Coordination between clauses* and *Right Word Location* independently, in addition to LEX_BASE, resulted in minor improvements of 0.2%, 0.15% and 0.45% respectively. *Left Word Location* feature did not result in any improvements. A combination of all dependency features has resulted in an improvement of 0.8% across all measurements of performance.

The significance of dependency features can be drawn from the impact on precision. Lexical features resulted in an improvement of 0.6% over baseline precision(88.26%) and the addition of dependency features further improved precision by 0.8%.

Performance reported by state of art systems for English are shown in Table 2. These have been trained on PDTB which has 40K discourse relations whereas our system is trained on HRDB which has 2K relations. We believe the performance of our system will improve as the size of HDRB increases.

Table 1: Performance of chosen Feature Set

Features Used	P	R	F	A
BASE=C-String	**88.26**	**62.5**	**73.05**	**90.12**
BASE+C-POS+C-POSN [-1,+1]	85.89	90.04	87.79	94.59
BASE+C-Chunk+C-ChunkN[-1,+1]	87.04	90.49	88.6	95.01
LEX_BASE	**88.88**	**91.51**	**90.05**	**95.66**
SYN_BASE + DEP_REL[1]	89.27	91.49	90.25	95.8
SYN_BASE +CON_COR[2]	89.0	91.64	90.19	95.73
SYN_BASE+R_WORD_LOC[3]	89.58	91.70	90.52	95.87
LEX_BASE+NEW[4]	**89.73**	**92.27**	**90.89**	**96.04**

[1]Dependency relation, [2]Coordination between clauses, [3]Right word location, [4]Dependency features

Table 2: Performance reported by systems for English

System	P	R	F	A
Pitler (2009)	-	-	94.19	96.26
Lin (2014)	-	-	95.36	97.25
Wang (2015)	95.28	95.00	95.14	-

4.2. Feature Set Comparison

To better understand the efficacy of our feature set, we report the performance of feature sets chosen by Pitler(2009) and Lin(2014) for Hindi in Table 3. However since Hindi TreeBank does not contain a constituent parse tree, “Parent category” feature from Pitler’s feature set and “path from node to root” feature from Lin’s feature set could not be implemented.

Table 3: Feature Set Comparision

System	P	R	F	A
Pitler (2009)	89.19	88.29	88.56	95.04
Lin (2014)	90.26	87.28	88.59	94.97
Our implementation	89.73	92.27	90.89	96.04

We report an increase of 2.3% in f-measure and 1% in accuracy over Pitler’s (2009) feature set and an increase of 2.3% in f-measure and 1.5% in accuracy over Lin’s (2014) feature set.

Carrying out such a feature set comparison across languages has led us to believe that a satisfactory baseline can be achieved using simple lexical features. Language specific features can be then explored depending upon additional information present (constituent parse trees for English and dependency parse trees for Hindi) and language specific differences with regards to what constitutes a discourse connective.

4.3. Other Experiments

To study the effect of dependency and lexical features separately, we experimented with combining models trained on different feature sets using standard linear interpolation:

$$P_{final}(C) = \lambda_1 * P_{f1}(C) + (1 - \lambda_1) * P_{f2}(C) \quad (1)$$

where C is the connective in question, P_{f1} is the model trained on lexical features, P_{f2} is the model trained on dependency features and λ_1 is the interpolation weight for P_{f1}. λ_1 was determined based on performance measured using ten-fold cross validation.

We also experimented with using connective specific models as was used by Elwell and Baldridge (2008) for discourse argument classification. Using connective specific models, we attempt to model each category of discourse connective independently to more closely reflect differing behaviors of each connective type. Discourse connectives behave differently according to the grammatical category they belong to. As a result we explore modelling each set independently As a result we trained specific models for each of the six categories of discourse connectives present in HDRB namely coordinating conjunctions, subordinating conjunctions, subordinators, sentential relatives, adverbials and particles. We then combine the connective specific models with a general model trained on all type of connectives:

$$P_{final}(C) = \lambda_2 * P_{conn}(C) + (1 - \lambda_2) * P_{gen}(C) \quad (2)$$

where C is the connective in question , P_{conn} is the connective specific model, P_{gen} is the general model and λ_2 is the interpolation weight for P_{conn}. λ_2 was determined based on performance measured using ten-fold cross validation.

We report that both experiments failed to improve over the performance of a single model trained on our chosen feature set.

5. Conclusions

In this work, we present the first component(discourse connective identifier) towards a fully automated discourse analyzer for Hindi. We have discussed in detail the performance of lexical and syntactic features such as C-String, C-POS, C-Chunk, C-POSN[+,-] and C-ChunkN[+,-] for this task, resulting in a f-measure of 90.9%. In addition to these we introduce 3 novel dependency features resulting in an further improvement of 0.8% across all measurements of performance.

6. Bibliographical References

Begum, R., Husain, S., Dhwaj, A., Sharma, D. M., Bai, L., and Sangal, R. (2008). Dependency annotation scheme for indian languages. In *IJCNLP*, pages 721–726. Citeseer.

Elwell, R. and Baldridge, J. (2008). Discourse connective argument identification with connective specific rankers. In *Semantic Computing, 2008 IEEE International Conference on*, pages 198–205. IEEE.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Kolachina, S., Prasad, R., Sharma, D. M., and Joshi, A. K. (2012). Evaluation of discourse relation annotation in the hindi discourse relation bank. In *LREC*, pages 823–828.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lin, Z., Ng, H. T., and Kan, M.-Y. (2014). A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Miltsakaki, E., Prasad, R., Joshi, A. K., and Webber, B. L. (2004). The penn discourse treebank. In *LREC*.

Mladová, L., Zikanova, S., and Hajicová, E. (2008). From sentence to discourse: Building an annotation scheme for discourse based on prague dependency treebank. In *LREC*.

Oza, U., Prasad, R., Kolachina, S., Sharma, D. M., and Joshi, A. (2009). The hindi discourse relation bank. In *Proceedings of the third linguistic annotation workshop*, pages 158–161. Association for Computational Linguistics.

Pitler, E. and Nenkova, A. (2009). Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16. Association for Computational Linguistics.

Poláková, L., Mírovskỳ, J., Nedoluzhko, A., Jínová, P., Zikánová, Š., and Hajičová, E. (2013). Introducing the prague discourse treebank 1.0. In *Proceedings of the 6th international joint conference on natural language processing*, pages 91–99.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008a). The penn discourse treebank 2.0. In *LREC*. Citeseer.

Prasad, R., Husain, S., Sharma, D. M., and Joshi, A. K. (2008b). Towards an annotated corpus of discourse relations in hindi. In *IJCNLP*, pages 73–80.

Wang, J. and Lan, M. (2015). A refined end-to-end discourse parser. *CoNLL 2015*, page 17.

Xue, N. (2005). Annotating discourse connectives in the chinese treebank. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 84–91. Association for Computational Linguistics.

Zeyrek, D. and Webber, B. L. (2008). A discourse resource for turkish: Annotating discourse connectives in the metu corpus. In *IJCNLP*, pages 65–72.

Zeyrek, D., Demirşahin, I., Sevdik-Çalli, A., Balaban, H. Ö., Yalçinkaya, İ., and Turan, Ü. D. (2010). The annotation scheme of the turkish discourse bank and an evaluation of inconsistent annotations. In *Proceedings of the fourth linguistic annotation workshop*, pages 282–289. Association for Computational Linguistics.

Zhang, H. (2004). The optimality of naive bayes. *AA*, 1(2):3.

Zhou, Y. and Xue, N. (2012). Pdtb-style discourse annotation of chinese text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 69–77. Association for Computational Linguistics.