Analysis of the Hindi Proposition Bank using Dependency Structure

Ashwini Vaidya Jinho D. Choi Martha Palmer Bhuvana Narasimhan
Institute of Cognitive Science
University of Colorado at Boulder
{vaidyaa,choijd,mpalmer,narasimb}@colorado.edu

Abstract

This paper makes two contributions. First, we describe the Hindi Proposition Bank that contains annotations of predicate argument structures of verb predicates. Unlike PropBanks in most other languages, the Hind PropBank is annotated on top of dependency structure, the Hindi Dependency Treebank. We explore the similarities between dependency and predicate argument structures, so the PropBank annotation can be faster and more accurate. Second, we present a probabilistic rule-based system that maps syntactic dependents to semantic arguments. With simple rules, we classify about 47% of the entire PropBank arguments with over 90% confidence. These preliminary results are promising; they show how well these two frameworks are correlated. This can also be used to speed up our annotations.

1 Introduction

Proposition Bank (from now on, PropBank) is a corpus in which the arguments of each verb predicate are annotated with their semantic roles (Palmer et al., 2005). PropBank annotation has been carried out in several languages; most of them are annotated on top of Penn Treebank style phrase structure (Xue and Palmer, 2003; Palmer et al., 2008). However, a different grammatical analysis has been used for the Hindi PropBank annotation, dependency structure, which may be particularly suited for the analysis of flexible word order languages such as Hindi.

As a syntactic corpus, we use the Hindi Dependency Treebank (Bhatt et al., 2009). Using dependency structure has some advantages. First, semantic arguments[1] can be marked explicitly on the syntactic trees, so annotations of the predicate argument structure can be more consistent with the dependency structure. Second, the Hindi Dependency Treebank provides a rich set of dependency relations that capture the syntactic-semantic information. This facilitates mappings between syntactic dependents and semantic arguments. A successful mapping would reduce the annotation effort, improve the inter-annotator agreement, and guide a full fledged semantic role labeling task.

In this paper, we briefly describe our annotation work on the Hindi PropBank, and suggest mappings between syntactic and semantic arguments based on linguistic intuitions. We also present a probabilistic rule-based system that uses three types of rules to arrive at mappings between syntactic and semantic arguments. Our experiments show some promising results; these mappings illustrate how well those two frameworks are correlated, and can also be used to speed up the PropBank annotation.

2 Description of the Hindi PropBank

2.1 Background

The Hindi PropBank is part of a multi-dimensional and multi-layered resource creation effort for the Hindi-Urdu language (Bhatt et al., 2009). This multi-layered corpus includes both dependency annotation as well as lexical semantic information in the form of PropBank. The corpus also produces phrase structure representations in addition to de-

[1]The term 'semantic argument' is used to indicate all numbered arguments as well as modifiers in PropBank.

pendency structure. The Hindi Dependency Treebank has created an annotation scheme for Hindi by adapting labels from Panini's Sanskrit grammar (also known as CPG: Computational Paninian Grammar; see Begum et al. (2008)). Previous work has demonstrated that the English PropBank tagset is quite similar to English dependency trees annotated with the Paninian labels (Vaidya et al., 2009). PropBank has also been mapped to other dependency schemes such as Functional Generative Description (Cinkova, 2006).

2.2 Hindi Dependency Treebank

The Hindi Dependency Treebank (HDT) includes morphological, part-of-speech and chunking information as well as dependency relations. These are represented in the Shakti Standard Format (SSF; see Bharati et al. (2007)). The dependency labels depict relations between chunks, which are "minimal phrases consisting of correlated, inseparable entities" (Bharati et al., 2006), so they are not necessarily individual words. The annotation of chunks also assumes that intra-chunk dependencies can be extracted automatically (Husain et al., 2010).

The dependency tagset consists of about 43 labels, which can be grouped into three categories: dependency relation labels, modifier labels, and labels for non-dependencies (Bharati et al., 2009). PropBank is mainly concerned with those labels depicting dependencies in the domain of locality of verb predicates. The dependency relation labels are based on the notion of 'karaka', defined as "the role played by a participant in an action". The karaka labels, k1-5, are centered around the verb's meaning. There are other labels such as rt (purpose) or k7t (location) that are independent of the verb's meaning.

2.3 Annotating the Hindi PropBank

The Hindi PropBank (HPB) contains the labeling of semantic roles, which are defined on a verb-by-verb basis. The description at the verb-specific level is fine-grained; e.g., 'hitter' and 'hittee'. These verb-specific roles are then grouped into broader categories using numbered arguments (ARG#). Each verb can also have modifiers not specific to the verb (ARGM*). The annotation process takes place in two stages: the creation of frameset files for individual verb types, and the annotation of predicate argument structures for each verb instance. As annotation tools, we use Cornerstone and Jubilee (Choi et al., 2010a; Choi et al., 2010b). The annotation is done on the HDT; following the dependency annotation, PropBank annotates each verb's syntactic dependents as their semantic arguments at the chunk level. Chunked trees are conveniently displayed for annotators in Jubilee. PropBank annotations generated in Jubilee can also be easily projected onto the SSF format of the original dependency trees.

The HPB currently consists of 24 labels including both numbered arguments and modifiers (Table 1). In certain respects, the HPB labels make some distinctions that are not made in some other language such as English. For instance, ARG2 is subdivided into labels with function tags, in order to avoid ARG2 from being semantically overloaded (Yi, 2007). ARGC and ARGA mark the arguments of morphological causatives in Hindi, which is different from the ARG0 notion of 'causer'. We also introduce two labels to represent the complex predicate constructions: ARGM-VLV and ARGM-PRX.

Label	Description		
ARG0	agent, causer, experiencer		
ARG1	patient, theme, undergoer		
ARG2	beneficiary		
ARG3	instrument		
ARG2-ATR	attribute	ARG2-GOL	goal
ARG2-LOC	location	ARG2-SOU	source
ARGC	causer		
ARGA	secondary causer		
ARGM-VLV	verb-verb construction		
ARGM-PRX	noun-verb construction[2]		
ARGM-ADV	adverb	ARGM-CAU	cause
ARGM-DIR	direction	ARGM-DIS	discourse
ARGM-EXT	extent	ARGM-LOC	location
ARGM-MNR	manner	ARGM-MNS	means
ARGM-MOD	modal	ARGM-NEG	negation
ARGM-PRP	purpose	ARGM-TMP	temporal

Table 1: Hindi PropBank labels.

2.4 Empty arguments in the Hindi PropBank

The HDT and HPB layers have different ways of handling empty categories (Bhatia et al., 2010). HPB inserts empty arguments such as PRO (empty subject of a non-finite clause), RELPRO (empty relative pronoun), pro (pro-drop argument), and gap-pro (gapped argument). HPB annotates syntactic relations between its semantic roles, notably co-indexation of the empty argument PRO as well as gap-pro. The example in Figure 1 shows that *Mohan* and PRO are co-indexed; thus, *Mohan* becomes ARG0 of *read* via the empty argument PRO. There is no dependency link between PRO and *read* because PRO is inserted only in the PropBank layer.

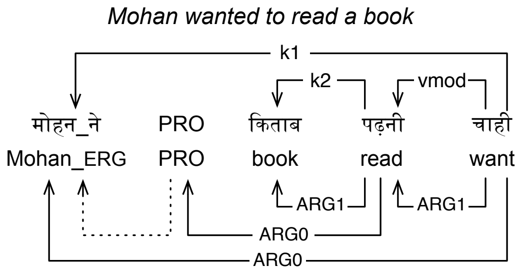

Figure 1: Empty argument example. The upper and lower edges indicate HDT and HPB labels, respectively.

3 Comparisons between syntactic and semantic arguments

In this section, we describe the mappings between HDT and HPB labels based on our linguistic intuitions. We show that there are several broad similarities between two tagsets. These mappings form the basis for our linguistically motivated rules in Section 4.2.3. In section 5.5, we analyze whether the intuitions discussed in this section are borne out by the results of our probabilistic rule-based system.

3.1 Numbered arguments

The numbered arguments correspond to ARG0-3, including function tags associated with ARG2. In PropBank, ARG0 and ARG1 are conceived as framework-independent labels, closely associated with Dowty's Proto-roles (Palmer et al., 2010). For instance, ARG0 corresponds to the agent, causer, or experiencer, whether it is realized as the subject of an active construction or as the object of an adjunct (by phrase) of the corresponding passive. In this respect, ARG0 and ARG1 are very similar to k1 and k2 in HDT, which are annotated based on their semantic roles, not their grammatical relation. On the other hand, HDT treats the following sentences similarly, whereas PropBank does not:

- The boy *broke* the window.
- The window *broke*.

The boy and *the window* are both considered k1 for HDT, whereas PropBank labels *the boy* as ARG0 and *The window* as ARG1. *The window* is not considered a primary causer as the verb is unaccusative for Propbank. For HDT, the notion of unaccusativity is not taken into consideration. This is an important distinction that needs to be considered while carrying out the mapping. k1 is thus ambiguous between ARG0 and ARG1. Also, HDT makes a distinction between Experiencer subjects of certain verbs, labeling them as k4a. As PropBank does not make such a distinction, k4a maps to ARG0. The Experiencer subject information is included in the corresponding frameset files of the verbs. The mappings to ARG0 and ARG1 would be accurate only if they make use of specific verb information. The mappings for other numbered arguments as well as ARGC and ARGA are given in Table 2.

HDT label	HPB label
k1 (karta); k4a (experiencer)	Arg0
k2 (karma)	Arg1
k4 (beneficiary)	Arg2
k1s (attribute)	Arg2-ATR
k5 (source)	Arg2-SOU
k2p (goal)	Arg2-GOL
k3 (instrument)	Arg3
mk1 (causer)	ArgC
pk1 (secondary causer)	ArgA

Table 2: Mappings to the HPB numbered arguments.

Note that in HDT annotation practice, k3 and k5 tend to be interpreted in a broad fashion such that they map not only to ARG3 and ARG2-SOU, but also to ARGM-MNS and ARGM-LOC (Vaidya and Husain, 2011). Hence, a one-to-one mapping for these labels is not possible. Furthermore, the occurrence of morphological causatives (ARGC and ARGA) is fairly low so that we may not be able to test the accuracy of these mappings with the current data.

3.2 Modifiers

The modifiers in PropBank are quite similar in their definitions to certain HDT labels. We expect a fairly high mapping accuracy, especially as these are not verb-specific. Table 3 shows mappings between

HDT labels and HPB modifiers. A problematic mapping could be ARGM-MNR, which is quite coarse-grained in PropBank, applying not only to adverbs of manner, but also to infinitival adjunct clauses.

HDT label	HPB label
sent-adv (epistemic adv)	ArgM-ADV
rh (cause/reason)	ArgM-CAU
rd (direction)	ArgM-DIR
rad (discourse)	ArgM-DIS
k7p (location)	ArgM-LOC
adv (manner adv)	ArgM-MNR
rt (purpose)	ArgM-PRP
k7t (time)	ArgM-TMP

Table 3: Mappings to the HPB modifiers.

3.3 Simple and complex predicates

HPB distinguishes annotations between simple and complex predicates. Simple predicates consist of only a single verb whereas complex predicates consist of a light verb and a pre-verbal element. The complex predicates are identified with a special label ARGM-PRX (ARGument-PRedicating eXpresstion), which is being used for all light verb annotations in PropBank (Hwang et al., 2010). Figure 2 shows an example of the predicating noun *mention* annotated as ARGM-PRX, used with *come*. The predicating noun also has its own argument, *matter of*, indicated with the HDT label r6-k1. The HDT has two labels, r6-k1 and r6-k2, for the arguments of the predicating noun. Hence, the argument span for complex predicates includes not only direct dependents of the verb but also dependents of the noun.

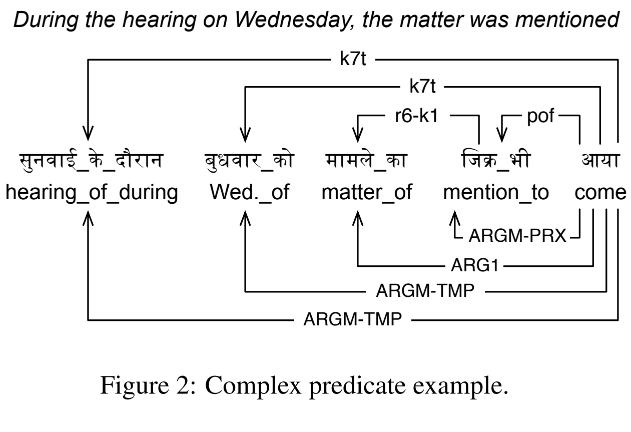

Figure 2: Complex predicate example.

The ARGM-PRX label usually overlaps with the HDT label pof, indicating a 'part of units' as pre-verbal elements in complex predicates. However, in certain cases, HPB has its own analysis for noun-verb complex predicates. Hence, not all the nominals labeled pof are labeled as ARGM-PRX. In the example in Figure 3, the noun chunk *important progress* is not considered to be an ARGM-PRX by HPB (in this example, we have *pragati hona*; (lit) progess be; to progress). The nominal for PropBank is in fact ARG1 of the verb *be*, rather than a composite on the verb. Additional evidence for this is that neither the nominal nor the light verb seem to project arguments of their own.

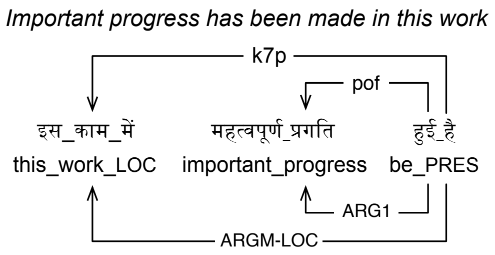

Figure 3: HDT vs. HPB on complex predicates.

4 Automatic mapping of HDT to HPB

Mapping between syntactic and semantic structures has been attempted in other languages. The Penn English and Chinese Treebanks consist of several semantic roles (e.g., locative, temporal) annotated on top of Penn Treebank style phrase structure (Marcus et al., 1994; Xue and Palmer, 2009). The Chinese PropBank specifies mappings between syntactic and semantic arguments in frameset files (e.g., SBJ → ARG0) that can be used for automatic mapping (Xue and Palmer, 2003). However, these Chinese mappings are limited to certain types of syntactic arguments (mostly subjects and objects). Moreover, semantic annotations on the Treebanks are done independently from PropBank annotations, which causes disagreement between the two structures.

Dependency structure transparently encodes relations between predicates and their arguments, which facilitates mappings between syntactic and semantic arguments. Hajičová and Kučerová (2002) tried to project PropBank semantic roles onto the Prague Dependency Treebank, and showed that the projection is not trivial. The same may be true to our case; however, our goal is not to achieve complete mappings between syntactic and semantic arguments, but to find a useful set of mappings that can speed up our annotation. These mappings will be applied to our future data as a pre-annotation stage, so that annotators do not need to annotate arguments that have already been automatically labeled by our system. Thus, it is important to find mappings with high precision and reasonably good recall.

In this section, we present a probabilistic rule-based system that identifies and classifies semantic arguments in the HPB using syntactic dependents in the HDT. This is still preliminary work; our system is expected to improve as we annotate more data and do more error analysis.

4.1 Argument identification

Identifying semantic arguments of each verb predicate is relatively easy given the dependency Treebank. For each verb predicate, we consider all syntactic dependents of the predicate as its semantic arguments (Figure 4). For complex predicates, we consider the syntactic dependents of both the verb and the predicating noun (cf. Section 3.3).

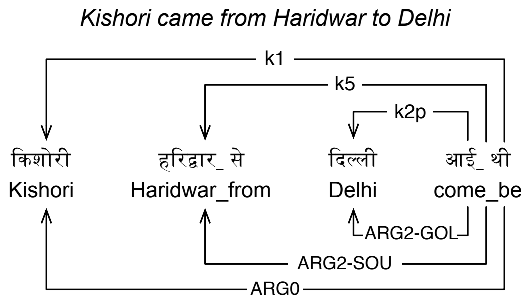

Figure 4: Simple predicate example.

With our heuristics, we get a precision of 99.11%, a recall of 95.50%, and an F1-score of 97.27% for argument identification. Such a high precision is expected as the annotation guidelines for HDT and HPB generally follow the same principles of identifying syntactic and semantic arguments of a verb. About 4.5% of semantic arguments are not identified by our method. Table 4 shows distributions of the most frequent non-identified arguments.

Label	Dist.	Label	Dist.	Label	Dist.
ARG0	3.21	ARG1	0.90	ARG2*	0.09

Table 4: Distributions of non-identified arguments caused by PropBank empty categories (in %).

Most of the non-identified argument are antecedents of PropBank empty arguments. As shown in Figure 1, the PropBank empty argument has no dependency link to the verb predicate. Identifying such arguments requires a task of empty category resolution, which will be explored as future work. Furthermore, we do not try to identify PropBank empty arguments for now, which will also be explored later.

4.2 Argument classification

Given the identified semantic arguments, we classify their semantic roles. Argument classification is done by using three types of rules. Deterministic rules are heuristics that are straightforward given dependency structure. Empirically-derived rules are generated by measuring statistics of dependency features in association with semantic roles. Finally, linguistically-motivated rules are derived from our linguistic intuitions. Each type of rule has its own strength; how to combine them is the art we need to explore.

4.2.1 Deterministic rule

Only one deterministic rule is used in our system. When an identified argument has a pof dependency relation with its predicate, we classify the argument as ARGM-PRX. This emphasizes the advantage of using our dependency structure: classifying ARGM-PRX cannot be done automatically in most other languages where there is no information provided for light verb constructions. This deterministic rule is applied before any other type of rule. Therefore, we do not generate further rules to classify the ARGM-PRX label.

4.2.2 Empirically-derived rules

Three kinds of features are used for the generation of empirically-derived rules: predicate ID, predicate's voice type, and argument's dependency label. The predicate ID is either the lemma or the roleset ID of the predicate. Predicate lemmas are already provided in HDT. When we use predicate lemmas, we assume no manual annotation of PropBank. Thus, rules generated from predicate lemmas can be applied to any future data without modification. When we use roleset ID's, we assume that sense annotations are already done. PropBank includes annotations of coarse verb senses, called roleset ID's, that differentiate each verb predicate with different senses (Palmer et al., 2005). A verb predicate can form several argument structures with respect to different senses. Using roleset ID's, we generate more fine-grained rules that are specific to those senses.

The predicate's voice type is either 'active' or 'passive', also provided in HDT. There are not many instances of passive construction in our current data, which makes it difficult to generate rules general enough for future data. However, even with the lack of training instances, we find some advantage of using the voice feature in our experiments. Finally, the argument's dependency label is the dependency label of an identified argument with respect to its predicate. This feature is straightforward for the case of simple predicates. For complex predicates, we use the dependency labels of arguments with respect to their syntactic heads, which can be pre-verbal elements. Note that rules generated with complex predicates contain slightly different features for predicate lemmas as well; instead of using predicate lemmas, we use joined tags of the predicate lemmas and the lemmas of pre-verbal elements.

ID	V	Drel	PBrel	#
come	a	`k1`	`ARG0`	1
come	a	`k5`	`ARG2-SOU`	1
come	a	`k2p`	`ARG2-GOL`	1
come_mention	a	`k7t`	`ARGM-TMP`	2
come_mention	a	`r6-k1`	`ARG1`	1

Table 5: Rules generated by the examples in Figures 4 and 2. The ID, V, and Drel columns show predicate ID, predicate's voice type, and argument's dependency label. The PBrel column shows the PropBank label of each argument. The # column shows the total count of each feature tuple being associated with the PropBank label. 'a' stands for active voice.

Table 5 shows a set of rules generated by the examples in Figures 4 (*come*) and 2 (*come_mention*). No rule is generated for `ARGM-PRX` because the label is already covered by our deterministic rule (Section 4.2.1). When roleset ID's are used in place of the predicate ID, *come* and *come_mention* are replaced with `A.03` and `A.01`, respectively. These rules can be formulated as a function $rule$ such that:

$$rule(id, v, drel) = \arg\max_i P(pbrel_i)$$

where $P(pbrel_i)$ is a probability of the predicted PropBank label $pbrel_i$, given a tuple of features $(id, v, drel)$. The probability is measured by estimating a maximum likelihood of each PropBank label being associated with the feature tuple. For example, a feature tuple (*come*, active, `k1`) can be associated with two PropBank labels, `ARG0` and `ARG1`, with counts of 8 and 2, respectively. In this case, the maximum likelihoods of `ARG0` and `ARG1` being associated with the feature tuple is 0.8 and 0.2; thus $rule$(*come*, active, `k1`) = `ARG0`.

Since we do not want to apply rules with low confidence, we set a threshold to $P(pbrel)$, so predictions with low probabilities can be filtered out. Finding the right threshold is a task of handling the precision/recall trade-off. For our experiments, we ran 10-fold cross-validation to find the best threshold.

4.2.3 Linguistically-motivated rules

Linguistically-motivated rules are applied to arguments that the deterministic rule and empirically-derived rules cannot classify. These rules capture general correlations between syntactic and semantic arguments for each predicate, so they are not as fine-grained as empirically-derived rules, but can be helpful for predicates not seen in the training data. The rules are manually generated by our annotators and specified in frameset files. Table 6 shows linguistically-motivated rules for the predicate 'A (*come*)', specified in the frameset file, 'A-v.xml'.[3]

Roleset	Usage	Rule

`A.01`	to come	`k1` $\rightarrow$ `ARG1`
		`k2p` $\rightarrow$ `ARG2-GOL`

`A.03`	to arrive	`k1` $\rightarrow$ `ARG1`
		`k2p` $\rightarrow$ `ARG2-GOL`
		`k5` $\rightarrow$ `ARG2-SOU`

`A.02`	light verb	No rule provided

Table 6: Rules for the predicate 'A (*come*)'.

The predicate 'A' has three verb senses and each sense specifies a different set of rules. For instance, the first rule of `A.01` maps a syntactic dependent with the dependency label `k1` to a semantic argument with the semantic label `ARG1`. Note that frameset files include rules only for numbered arguments. Most of these rules should already be included in the empirically-derived rules as we gain

[3]See Choi et al. (2010a) for details about frameset files.

more training data; however, for an early stage of annotation, these rules provide useful information.

5 Experiments

5.1 Corpus

All our experiments use a subset of the Hindi Dependency Treebank, distributed by the ICON'10 contest (Husain et al., 2010). Our corpus contains about 32,300 word tokens and 2,005 verb predicates, in which 546 of them are complex predicates. Each verb predicate is annotated with a verse sense specified in its corresponding frameset file. There are 160 frameset files created for the verb predicates. The number may seem small compared to the number of verb predicates. This is because we do not create separate frameset files for light verb constructions, which comprise about 27% of the predicate instances (see the example in Table 6).

All verb predicates are annotated with argument structures using PropBank labels. A total of 5,375 arguments are annotated. Since there is a relatively small set of data, we do not make a separate set for evaluations. Instead, we run 10-fold cross-validation to evaluate our rule-based system.

5.2 Evaluation of deterministic rule

First, we evaluate how well our deterministic rule classifies the ARGM−PRX label. Using the deterministic rule, we get a 94.46% precision and a 100% recall on ARGM−PRX. The 100% recall is expected; the precision implies that about 5.5% of the time, light verb annotations in the HPB do not agree with the complex predicate annotations (pof relation) in the HDT (cf. Section 3.3). More analysis needs to be done to improve the precision of this rule.

5.3 Evaluation of empirically-derived rules

Next, we evaluate our empirically-derived rules with respect to the different thresholds set for $P(pbrel_i)$. In general, the higher the threshold is, the higher and lower the precision and recall become, respectively. Figure 5 shows comparisons between precision and recall with respect to different thresholds. Notice that a threshold of 1.0, meaning that using only rules with 100% confidence, does not give the highest precision. This is because the model with this high of a threshold overfits to the training data. Rules that work well in the training data do not necessarily work as well on the test data.

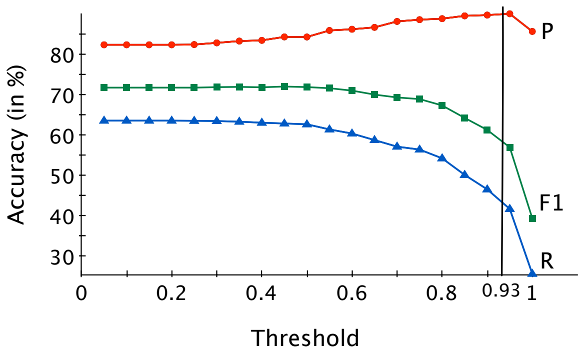

Figure 5: Accuracies achieved by the empirically derived rules using (lemma, voice, label) features. P, R, and F1 stand for precisions, recalls, and F1-scores, respectively.

We need to find a threshold that gives a high precision (so annotators do not get confused by the automatic output) while maintaining a good recall (so annotations can go faster). With a threshold of 0.93 using features (lemma, voice, dependency label), we get a precision of 90.37%, a recall of 44.52%, and an F1-score of 59.65%. Table 7 shows accuracies for all PropBank labels achieved by a threshold of 0.92 using roleset ID's instead of predicate's lemmas. Although the overall precision stays about the same, we get a noticeable improvement in the overall recall using roleset ID's. Note that some labels are missing in Table 7. This is because either they do not occur in our current data (ARGC and ARGA) or we have not started annotating them properly yet (ARGM−MOD and ARGM−NEG).

5.4 Evaluation of linguistically-motivated rules

Finally, we evaluate the impact of the linguistically-motivated rules. Table 8 shows accuracies achieved by the linguistically motivated rules applied after the empirically derived rules. As expected, the linguistically motivated rules improve the recall of ARGN significantly, but bring a slight decrease in the precision. This shows that our linguistic intuitions are generally on the right track. We may combine some of the empirically derived rules with linguistically motivated rules together in the frameset files so annotators can take advantage of both kinds of rules in the future.

	Dist.	P	R	F1
ALL	100.00	90.59	47.92	62.69
ARG0	17.50	95.83	67.27	79.05
ARG1	27.28	94.47	61.62	74.59
ARG2	3.42	81.48	37.93	51.76
ARG2-ATR	2.54	94.55	40.31	56.52
ARG2-GOL	1.61	64.29	21.95	32.73
ARG2-LOC	0.87	90.91	22.73	36.36
ARG2-SOU	0.83	78.26	42.86	55.38
ARG3	0.08	0.00	0.00	0.00
ARGM-ADV	3.50	31.82	3.93	7.00
ARGM-CAU	1.44	50.00	5.48	9.88
ARGM-DIR	0.43	100.00	18.18	30.77
ARGM-DIS	1.63	26.67	4.82	8.16
ARGM-EXT	1.42	0.00	0.00	0.00
ARGM-LOC	10.77	83.80	27.42	41.32
ARGM-MNR	6.00	57.14	9.18	15.82
ARGM-MNS	0.79	77.78	17.50	28.57
ARGM-PRP	2.15	65.52	17.43	27.54
ARGM-PRX	10.75	94.46	100.00	97.15
ARGM-TMP	7.01	74.63	14.04	23.64

Table 7: Labeling accuracies achieved by the empirically derived rules using (roleset ID, voice, label) features and a threshold of 0.92. The accuracy for ARGM-PRX is achieved by the deterministic rule. The Dist. column shows a distribution of each label.

	Dist.	P	R	F1
ALL	100.00	89.80	55.28	68.44
ARGN	54.12	91.87	72.36	80.96
ARGM	45.88	85.31	35.14	49.77
ARGN w/o LM		93.63	58.76	72.21

Table 8: Labeling accuracies achieved by the linguistically motivated rules. The ARGN and ARGM rows show statistics of all numbered arguments and modifiers combined, respectively. The 'ARGN w/o LM' row shows accuracies of ARGN achieved only by the empirically derived rules.

5.5 Error anlaysis

The precision and recall results for ARG0 and ARG1, are better than expected, despite the complexity of the mapping (Section 3.1). This is because they occur most often in the corpus, so enough rules can be extracted. The other numbered arguments are closely related to particular types of verbs (e.g., motion verbs for ARG2-GOL|SOU). Our linguistically motivated rules are more effective for these types of HPB labels. We would expect the modifiers to be mapped independently of the verb, but our experiments show that the presence of the verb lemma feature enhances the performance of modifiers. Although section 3.2 expects one-to-one mappings for modifiers, it is not the case in practice.

We observe that the interpretation of labels in annotation practice is important. For example, our system performs poorly for ARGM-ADV because the label is used for various sentential modifiers and can be mapped to as many as four HDT labels. On the other hand, HPB makes some fine-grained distinctions. For instance, means and causes are distinguished using ARGM-CAU and ARGM-MNS labels, a distinction that HDT does not make. In the example in Figure 6, we find that *aptitude_with* is assigned to ARGM-MNS, but gets the cause label rh in HDT.

Rajyapal can call upon any party with his aptitude

राज्यपाल	अपने	विवेक_से	किसी_भी	पार्टी_को	बुला_सकता_है
Rajyapal	his	aptitude_with	any_EMPH	party_DAT	call_can_be

Figure 6: Means vs. cause example.

6 Conclusion and future work

We provide an analysis of the Hindi PropBank annotated on the Hindi Dependency Treebank. There is an interesting correlation between dependency and predicate argument structures. By analyzing the similarities between the two structures, we find rules that can be used for automatic mapping of syntactic and semantic arguments, and achieve over 90% confidence for almost half of the data. These rules will be applied to our future data, which will make the annotation faster and possibly more accurate.

We plan to use different sets of rules generated by different thresholds to see which rule set leads to the most effective annotation. We also plan to develop a statistical semantic role labeling system in Hindi, once we have enough training data. In addition, we will explore the possibility of using existing lexical resource such as WordNet (Narayan et al., 2002) to improve our system.

Acknowledgements

This work is supported by NSF grants CNS- 0751089, CNS-0751171, CNS-0751202, and CNS-0751213. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *In Proceedings of the 3rd International Joint Conference on Natural Language Processing*, IJCNLP'08.

Akshar Bharati, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2006. AnnCorra: Guidelines for POS and Chunk Annotation for Indian Languages. Technical report, IIIT Hyderabad.

Akshar Bharati, Rajeev Sangal, and Dipti Misra Sharma. 2007. Ssf: Shakti standard format guide. Technical report, IIIT Hyderabad.

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

Archna Bhatia, Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Michael Tepper, Ashwini Vaidya, and Fei Xia. 2010. Empty categories in a hindi treebank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, pages 1863–1870.

Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. In *In the Proceedings of the Third Linguistic Annotation Workshop held in conjunction with ACL-IJCNLP 2009*.

Jinho D. Choi, Claire Bonial, and Martha Palmer. 2010a. Propbank frameset annotation guidelines using a dedicated editor, cornerstone. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, LREC'10, pages 3650–3653.

Jinho D. Choi, Claire Bonial, and Martha Palmer. 2010b. Propbank instance annotation guidelines using a dedicated editor, jubilee. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, LREC'10, pages 1871–1875.

Silvie Cinkova. 2006. From PropBank to EngVALLEX: Adapting PropBank-Lexicon to the Valency Theory of Functional Generative Description. In *Proceedings of the fifth International conference on Language Resources and Evaluation (LREC 2006), Genova, Italy*.

Eva Hajičová and Ivona Kučerová. 2002. Argument/valency structure in propbank, lcs database and prague dependency treebank: A comparative pilot study. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, LREC'02, pages 846–851.

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

Jena D. Hwang, Archna Bhatia, Claire Bonial, Aous Mansouri, Ashwini Vaidya, Nianwen Xue, and Martha Palmer. 2010. PropBank Annotation of Multilingual Light Verb Constructions. In *Proceedings of the Linguistic Annotation Workshop at ACL 2010*.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*, pages 114–119.

Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. An experience in building the indo wordnet - a wordnet for hindi. In *Proceedings of the 1st International Conference on Global WordNet*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot arabic propbank. In *Proceedings of the 6th International Language Resources and Evaluation*, LREC'08, pages 28–30.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. In Graeme Hirst, editor, *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool.

Ashwini Vaidya and Samar Husain. 2011. A classification of dependencies in the Hindi/Urdu Treebank. In *Presented at the Workshop on South Asian Syntax and Semantics, Amherst, MA*.

Ashwini Vaidya, Samar Husain, and Prashanth Mannem. 2009. A karaka based dependency scheme for English. In *Proceedings of the CICLing-2009, Mexico City, Mexico*.

Nianwen Xue and Martha Palmer. 2003. Annotating the propositions in the penn chinese treebank. In *Proceedings of the 2nd SIGHAN workshop on Chinese language processing*, SIGHAN'03, pages 47–54.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15(1):143–172.

Szu-Ting Yi. 2007. *Automatic Semantic Role Labeling*. Ph.D. thesis, University of Pennsylvania.