Spring 1-1-2015

Hindi Complex Predicates: Linguistic and Computational Approaches

Ashwini Vaidya
University of Colorado at Boulder, ashwini.vaidya@gmail.com

Hindi Complex Predicates: Linguistic and Computational Approaches

by

Ashwini Vaidya

M.A., University of Mumbai, 2005

M.Phil., IIIT Hyderabad, 2009

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Linguistics

2015

This thesis entitled:
Hindi Complex Predicates: Linguistic and Computational Approaches
written by Ashwini Vaidya
has been approved for the Department of Linguistics

_____

Dr. Martha Palmer

_____

Dr. Bhuvana Narasimhan

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Vaidya, Ashwini (Ph.D., Linguistics and Cognitive Science)

Hindi Complex Predicates: Linguistic and Computational Approaches

Thesis directed by Dr. Martha Palmer and Dr. Bhuvana Narasimhan

Complex predicates that comprise of a noun and verb e.g. *yaad kar* 'memory do; remember' are a productive class of multi-words in Hindi. In this thesis, we examine the challenges of identification and representation for these complex predicates in Hindi. We design and implement their representation in a lexical semantic resource as well as in lexicalized computational grammars. As productive multi-word predicates, their accurate identification is a necessity for natural language processing applications. We use a combination of linguistic and computational approaches to address these challenges. We use these methods to demonstrate the semi-automatic creation of subcategorization frames for Hindi and the development of classes for nominal predicates. Finally, we demonstrate how linguistic features and computational tools can be used in tandem to automatically identify complex predicates from unseen text.

Dedication

To my grandparents.

Acknowledgements

I gratefully acknowledge my co-advisors Martha Palmer and Bhuvana Narasimhan. Martha's support came in many forms—as an academic advisor, an experienced computational linguist and perhaps most importantly as an optimist. After a discussion about a difficult problem, I usually left her office with a positive feeling that encouraged me to work towards a solution. I enjoyed my discussions with Bhuvana on various aspects of Hindi. Her incisive comments always improved my writing in many ways, and made me think about problems more deeply. Bhuvana also encouraged me to be thorough about all aspects of research and I will always be thankful to her for that.

Others at CU Linguistics, ICS and beyond were also supportive and I thank Laura Michaelis, Jim Martin and Owen Rambow for their feedback on this dissertation. A little later during my PhD, I spent some time at the University of Konstanz in Miriam Butt's group. I consider myself lucky to have had an opportunity to discuss my work with her and her feedback has improved this dissertation considerably.

I thank my linguistics colleagues in Boulder and Konstanz—Jinho Choi, Archna Bhatia, Annette Hautli, Jena Hwang, Aous Mansouri, Sebastian Sulger and Shumin Wu. I learnt a lot from discussions with them. I thank my friends in Boulder, especially Ashmi Desai at whose home I enjoyed many long chats over chai; Adi Renduchintala for the beer and the films at IFS and Sam and Jena for being such great roommates. The support of my family–my parents, my late grandparents and my brother meant the world to me over the years. Finally, I thank Samar for his unwavering belief in me as we navigated two dissertations and a post-doc across three countries.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Overview

The representation and identification of multi-word expressions has been one of the major challenges for both linguistic representation and computational processing. Multi-words are very broadly defined as "idiosyncratic interpretations that cross word boundaries or spaces" (Sag et al., 2002). This definition assumes two criteria for multi-words, first that they have a special semantics and second, that they span across one or more words in a sentence. Although there are a plethora of multi-words in language (ranging from compounds to idioms), those multi-words that also occur as predicates are especially relevant to this work.

Multi-word predicates are a particularly challenging subset of multi-word expressions. Although they behave as a single predicating expression, they are composed of more than one lexical unit. In the languages of South Asia, multi-word predicates, which are more commonly known as complex predicates, are particularly pervasive (Masica, 1976). Consequently, complex predicates have been a part of linguistic analyses and descriptive grammars for a long time (Bahl, 1974; Hook, 1974). The term 'complex predicate' is also a cover term for several sub-types of multi-word predicates including morphological causatives, noun-verb complex predicates and verb-verb complex predicates. In this thesis, we focus on noun-verb complex predicates in Hindi.

1.2 Hindi

Hindi belongs to the Indo-Aryan branch of the Indo-Iranian languages. It is an SOV (verb-final) language with relatively free word order. It is a split-ergative language, where the ergative pattern is conditioned by the perfective tense on the verb. Usually, agents of transitive or ditransitive verbs will get ergative case if the verb has perfective aspect (although there are intransitive verbs e.g. *chiinkh* 'sneeze' that can appear with ergative case).

The rules for Hindi verbal agreement are also different from a language like English. For instance, the verb will always agree with the highest nominative (null) marked argument, which is not necessarily the syntactic subject. In simple verbs, in examples (1) and (2), the nominative argument is the subject and the object respectively. In example (3), there is no nominative argument available, thus the verb shows default agreement with third person, masculine, singular. In complex predicates, the light verb can show agreement with the nominal (with some important exceptions), provided no other argument in the sentence has nominative case (example (4)).

(1) laṛkii laṛke=ko dekʰ-egii
girl.F.Sg.Nom boy.M.Sg=Acc see-fut.F.Sg
'(The) girl will see (the) boy'

(2) laṛke=ne capaaṯ̃i kʰaa-ii
boy.M.Sg=Erg bread.F.Sg.Nom eat-Perf.F.Sg
'(The) boy ate (the) bread'

(3) laṛkii=ne auraṯ̃=ko dekʰ-aa
girl.F.Sg=Erg woman.F.Sg=Acc see-Perf.M.Sg
'(The) girl saw (the) woman'

(4) laṛkii=ne laṛke=se nafraṯ̃ k-ii
girl.F.Sg=Erg boy.M.Sg=Instr hatred.F do-Perf.F.Sg
'(The) girl hated (the) boy'

Hindi also regularly pro-drops arguments when they can be recovered from discourse or the particular situational context. Example 5 shows a sentence without a subject argument.

(5) Elided subject represented with *pro*:

pro kiṯaab paṛRh-egii
NULL book.F.Sg read-Fut.F.Sg

'(She) will read the book'.

The variety of Hindi used in this thesis is modern standard Hindi. At a few points in the thesis, we also make reference to Urdu, especially with regard to the Lexical Functional Grammar analysis for Urdu. We would like to note that Urdu and Hindi are syntactically very similar, but have some differences in vocabulary—mainly with respect to nouns. Urdu contains nouns that are of Persian and Arabic origin in contrast to Hindi, where the nouns are often of Sanskrit origin.

1.3 Noun-Verb Complex Predicates in Hindi

The term complex predicate can be defined as a "construction in which two or more predicational elements each contribute to a joint predication"(Butt, 2010). In the case of Hindi, the two predicational elements can be morphological (e.g. morphological causatives) or syntactic (noun-verb or verb-verb complex predicates) in nature. Syntactic complex predicates consist of a verbal element that combines with another pre-verbal element. In Hindi, the latter may include another verb, an adjective, an adverb, a borrowed English verb or a noun. Examples 6-8 contrast the use of a simple predicate *de* 'give' with its complex predicate usages.

(6) Simple predicate

raam=ne mohan=ko kiṯab ḏ-ii
Ram.M.Sg=Erg Mohan.M.Sg=Dat book.F.Sg give-Perf.F.Sg

'Ram gave Mohan a book'

(7) Noun-Verb complex predicate

raam=ne us baaṯ=par **zor** **ḏi-yaa**
Ram.M.Sg=Erg that topic=loc pressure.M.Sg give-Perf.M.Sg

'Ram put an emphasis on that topic'

(8) Verb-Verb complex predicate

raam=ne mohan=ko peṛ **kaatne ḍi-yaa**
Ram.M.Sg=Erg Mohan.M.Sg=acc tree.M.Sg cut.Inf give-Perf.M.Sg

'Ram let Mohan cut the tree'

In examples (7) and (8), verb *de* 'give' has depleted predicational power and requires an additional preverbal element to complete the event description. Therefore, it is often referred to as a *light verb*. Jespersen (1965) was the first to coin the term *light verb* with reference to English constructions such as *take a walk* or *give a sigh*. An example such as (7) is a combination of a *noun* and a *light verb*. In this work, we will use the abbreviation Noun-Verb Complex predicate (NVC) to refer to examples such as 7. We prefer to use 'noun-verb complex predicate' rather than the more ambiguous 'light verb construction' as the light verb may combine with either verb or noun in Hindi (see examples 7 and 8). NVCs have also been termed as, 'support verb' or 'conjunct verb' constructions.

NVCs are found across languages e.g. Japanese, Korean, Persian as well as English. In Hindi, NVCs are a productive class of multi-word predicates and present descriptive and theoretical challenges to the linguist. From the point of view of computational linguistics, NVCs pose challenges for lexical resource creation and require reliable diagnostic criteria to distinguish them from literal i.e. non-NVC cases. A deeper understanding of NVCs will benefit both linguistic analysis and computational applications. In the rest of the chapter, we examine some of these challenges in detail and outline our approach towards them.

1.4 Productivity of NVCs in Hindi

NVCs form a large part of the lexicon in Hindi and are highly productive. In the Hindi treebank (Palmer et al., 2009) (400,000 words), there are nearly 47,163 predicates, of which 37% have been annotated as NVCs. In comparison, English NVCs are considerably fewer. Linguistic analyses of Hindi NVCs are present in the literature from early work such as Kachru (1982) to more recent work e.g. Mohanan (1994); Davison (2005); Ahmed and Butt (2011). Most of this work is

based on NVC data from a small number of examples. Fewer studies have used corpora (although Bahl (1974) represents an early corpus study).

One of the challenges of studying Hindi NVCs stems from a wide range of noun-light verb combinations. Predicating nouns can combine with more than one light verb, resulting in potentially many different NVC combinations. These combinations are not completely predictable, as it is possible to find cases where a noun is incompatible with a particular light verb. In the examples below, we find that certain combinations are ruled out as ungrammatical.

- $b^haas^ha\dot{n}$ 'speech' + *kar* 'do'/ *de* 'give'/**le* 'take'
- *salaaha* 'advice' + **kar* 'do'/*de* 'give' /*le* 'take'
- d^hamkii 'threat' + **kar* 'do'/*de* 'give' /**le* 'take'

There is a need to investigate the properties of NVCs on a larger scale by making use of existing linguistic resources for Hindi. This would help us uncover generalizations or patterns that could explain and possibly predict NVC cases. In particular, it would be useful to understand the properties of nominal predicates further. We know that light verbs viz. *do*, *give*, *make* are a relatively compact class of light verbs that appear cross-linguistically. However, the properties of the nouns and their constraints on combination with light verbs are less well understood.

Nouns that appear as part of NVCs are also not as predictable as those found in in other languages. For example, in English, NVCs such as *make an offer*, *give a groan* often combine a nominalized form of an English verb e.g. *offer* or *groan* with a light verb. Consequently, a light verb construction may be paraphrased by the verbal form of the noun in English e.g. *gave a lecture* may be paraphrased by *lectured*. In contrast, nouns that occur as part of Hindi NVCs are very rarely nominalizations of main verbs. An example of such an exceptional case is *talaash kar* 'search do; search', where the noun also occurs as a main verb *talaashnaa* 'to search' (Davison, 2005).

Butt (2010) notes that light verbs in Hindi NVCs act as a verbalizers in order to create new predicates and incorporate borrowed items into the language e.g. *email kar* 'email do; email'.

Therefore, NVCs are sometimes described as "a preferred way of augmenting the creative potential of the language" (Kachru, 2006)[93]. It is not uncommon to find examples of borrowed words among NVCs in other languages e.g. Korean complex predicates will borrow nouns of Chinese origin to form NVCs (Han and Rambow, 2000).

The productivity of NVCs results in predicate types that are far more numerous than simple verbs. Hindi has approximately 700 simple verbs, but potentially many more unique NVCs[1]. While corpus data containing a range of NVCs was limited in availability earlier, in this thesis we take advantage of the annotated Hindi Treebank for the study of NVCs.

1.5 Identification of NVCs

As the Hindi Treebank annotations are our primary source of information about NVCs, we review their criteria to distinguish NVCs from other verb-argument combinations. NVCs appear to lie on a continuum between literal noun and verb combinations and prototypical NVC cases. Therefore, some linguistic diagnostic tests can predict a prototypical case of an NVC, but cannot work as predicted for a borderline NVC case. An extensive review of linguistic diagnostic tests from the existing literature was carried out before the annotation of NVCs in the Hindi Treebank. Mohanan (1994), Bhattacharyya et al. (2007) and Ahmed et al. (2012) have explored linguistic tests that are aimed at showing that the noun and light verb combinations form a complex predicate rather than an ordinary predicate argument structure. Some of these include addition of accusative case clitic, relativization and co-ordination.

Ahmed et al. (2012) describe the extra argument test for NVCs with transitive light verbs. In contrast with simple transitive clauses where the noun does not contribute any additional arguments, in an NVC the noun contributes an argument of its own. In the Hindi Treebank, the criteria taken into account for annotating a case of an NVC is based mainly on the existence of an extra argument. Examples such as 9 will not be annotated as NVCs in the Treebank as the noun

[1] http://verbs.colorado.edu/propbank/framesets-hindi/

kaam ‘work’ does not contribute arguments of its own.

(9) *raam=ne* *kaam* *ki-yaa*
Ram.M.Sg=Erg work.M do-perf.M.Sg
‘Ram did his work’

As the extra argument test can only be applied to transitive complex predicates, other tests such as the addition of the reflexive possessive *apnaa-apna*; ‘self-self’ are used for intransitive complex predicates. In the following chapter, we review some of these tests in more detail.

1.5.1 Automatic Identification

Based on human disambiguation of NVC cases from non-NVC ones, it is possible to come up with classifiers that carry out this task automatically for a large number of NVCs. Effective NVC identification has been shown to improve parsing (Begum et al., 2011), word sense disambiguation (Finlayson and Kulkarni, 2011) and machine translation (Pal et al., 2011). The linguistic diagnostic features described in the earlier section need to be modified and used as features for automatic identification. Previous work has underscored the importance of linguistic features in identifying NVCs automatically, across languages (Tu and Roth, 2011; Vincze et al., 2011). NVC identification can also be described in terms of disambiguating light and non-light usages of a verb. Therefore, an effective NVC identification system should not only use the best combination of linguistic features, but it should also be able to identify a range of light verb usages.

In Hindi, just as all full, lexical verbs exhibit differences in syntactic behaviour, all light verbs are not alike. Most work has focused on the most frequently occurring light verb *kar*, but there is some evidence that light verbs can be differentiated amongst themselves on the basis of the semantic information that they contribute to the NVC. For example, light verbs *kar* ‘do’ and *ho* ‘be’ contribute less semantic information than light verbs like *de* ‘give’ or *le* ‘take’ (Ahmed, 2010). Evaluation by light verb can reveal the robustness of the NVC identification system and can also provide a linguistically motivated guideline for improvement.

1.6 Challenges in Linguistic Representation

The representation of NVCs in grammars and linguistic resources can be a challenging task, given the multi-component nature of these predicates as well as their productivity. Broadly, there are two issues that need to be addressed. The first is related to the representation of multi-word predicates in lexicalized grammars, and the second is about the pragmatic or implementational aspects of linguistic resource creation.

1.6.1 Lexicalized Grammars

In the Hindi Treebank, NVCs are differentiated from ordinary arguments with the help of a dependency label. One of the challenges is to investigate whether there are syntactic representations apart from the more under-specified dependency representation that can capture the properties of Hindi NVCs more thoroughly. Lexicalized grammar formalisms have been used successfully to model NVCs (Abeillé, 1988; Han and Rambow, 2000; Müller, 2010; Ahmed et al., 2012). The earliest analyses assume that the light verb inherits arguments from the nominal. Its only function is to supply verbal case to the semantic arguments of the nominal and give up all its predicating power (Grimshaw and Mester, 1988; Kearns, 1988). Such an analysis requires a transfer of arguments from the nominal to the light verb. In a HPSG (Head-driven Phrase Structure Grammar) analysis of Persian NVCs, a similar analysis is described, via the lexical entries of the light verb and nominal. The noun is specified for all the arguments of the light verb and subsequently, the light verb 'attracts' the arguments of the noun (Müller, 2010).

Alternatively, it has also been proposed that the noun in the NVC behaves similarly to a verb in a control construction (Huang, 1992). NVCs consist of a light verb in the matrix clause with a nominal host as its complement. In comparison to the Grimshaw and Mester (1988) analysis, the control analysis explains the problem of 'transfer of arguments' more easily. Nominal hosts act as the head of a complement clause, without any expressed subject. The control approach, however, does not take into account that complex predicates in Hindi are monoclausal rather than having

an embedded, biclausal structure (Mohanan, 1997).

Finally, the work of Mohanan (1994); Alsina et al. (1997) and Butt (1995), focus on the joint predication of noun and light verb, where both parts of the predication contribute their meaning. Rather than view the light verb as a licenser of predication only, Butt (1993) views it as a unique category that shares a lexical entry with its non-light form. It is resolved as light or non-light depending upon the syntactic environment. When a verb is resolved as light, it supplies additional aspectual or agentive information about the event, and the host provides additional eventive meanings. These syntactic and semantic interactions with the host change the event structure of the ordinary predicate into a complex predicate.

This review of previous work suggests that there are two broad trends with respect to the representation of NVCs: one that considers the light verb as semantically empty, and one that does not. With respect to the Hindi NVC, one of the challenges is to compare these contrasting views in order to understand which approach might work best for the NVC.

1.6.2 Resource creation

A linguistic representation of NVCs must take into account the language specific properties of NVCs. Cross-linguistically, NVCs have a few similarities but differ in a number of ways. In Persian, there are only 250 simple verbs and almost all other predicates in use are NVCs (Sadeghi, 1993). In English, NVCs exist alongside other multi-word predicates like phrasal verbs as well as simple verbs. In Hindi, NVCs are a highly productive predicative strategy, but this productivity is confined to a small group of light verbs, specifically *kar* 'do' and *ho* 'be'. In comparison, light verbs like *le* 'take' or *aa* 'come' are limited with respect to their combinatorial possibilities. Therefore, resource creation e.g. in Hindi WordNet (Bhattacharyya, 2010) tends to focus on the light verb *kar* 'do' only.

For tasks like semantic role labelling, where every instance of a multi-word has to be annotated with its appropriate roles, NVCs need to be listed out in order to obtain accurate annotations for these cases at a sentential level. At the same time, maintaining consistency across annotations is

often a difficult task, especially when there are many unique NVC cases in the data. At such times, it is useful to have some over-arching generalizations about nouns that behave in a similar fashion. In this way, nouns belonging to the same group or class can be analyzed consistently. Therefore, the lexical resource creation problem must deal with both adequate coverage for the purpose of building a useful resource, as well as generalizability in order to maintain consistency.

1.7 Approach

So far, we have identified empirical, representational and identification challenges with respect to Hindi NVCs. In this thesis, our aim is to address these issues using a combination of linguistic analysis and computational tools. Our approach makes use of the Hindi Treebank, consisting of syntactic and semantic annotations (Palmer et al., 2009). This particular resource has 400,000 words and nearly a third of all the predicates in this Treebank are annotated as NVCs.

As a first step, we manually validate Treebank annotations of NVCs using linguistic diagnostic tests. Following this, we examine the problem of lexical resource creation for NVCs. For the purpose of semantic role annotation, we need to define semantic roles for every NVC in the Treebank in the form of 'frame files'. As the manual creation of these frame files is difficult due to the large number of NVC combinations, we implement a semi-automatic method for generating semantic roles from the informative dependency labels. The roles are then manually validated to result in semantic frame definitions for nearly 1800 nominal predicates.

While this directly aids us in the annotation task, we also carry out an initial exploration of noun classes for nominal predicates. This would help us in grouping together similar nouns and improving consistency in annotation. We carry out a clustering analysis on the nouns in the Hindi treebank using two linguistic features and use some of our results to improve coverage for a Hindi/Urdu LFG grammar.

We continue our investigation of linguistic representation of NVCs in the next stage, where we turn to two grammar formalisms: Tree Adjoining Grammar (TAG) and Lexical Functional Grammar (LFG). While an analysis of NVCs already exists in LFG, we adapt a TAG analysis

of NVCs for Hindi. We also compare the representation of NVCs in both formalisms to arrive at a clearer understanding of the formalisms and the syntactic propertis of the NVC itself. An additional motivation is the possibility of using the existing Treebank representation to extract a TAG or LFG representation of NVCs, which can aid an NLP application like parsing.

Finally, we examine the challenges of identifying NVCs with accuracy by building an automatic identification system that uses both linguistic and statistical features. Our model predicts unseen NVCs with an accuracy of upto 88% on a within-genre test set and 86% on a test set from a different genre. We utilize features based on our linguistic analysis and additionally evaluate our system on individual light verbs. Our model shows some improvement with compared with previous work on Hindi NVC identification.

We have shown that Hindi NVCs present a number of theoretical and empirical challenges for the linguist. In the thesis, we have organized our approach in the following manner. In Chapter 2 we provide an overview of the language resources used in this thesis. Next, we describe our lexical semantic representation for NVCs in Chapter 3 and the clustering technique for finding noun groups in Chapter 4. The syntactic representation of NVCs using Lexical Functional Grammar and Tree-Adjoining Grammar is examined in chapter 5. Finally, we discuss the results of the automatic identification of NVCs in Chapter 6 and follow up with a conclusion and discussion about future work.

Chapter 2

Language resources

In this chapter, we elaborate on some existing linguistic resources for Hindi, which we utilize for the study of Hindi NVCs. These resources are used to obtain linguistic knowledge about complex predicates. We will describe the structure and use of these resources in the following sections.

2.1 Hindi Treebank

The Hindi Treebank is a part of the Hindi and Urdu Treebank project. This is a multi-dimensional and multi-layered resource creation effort for the Hindi and Urdu languages (Palmer et al., 2009). It has two layers of annotation: dependency structure as well as lexical semantic information. As a multi-dimensional treebank, it includes both dependency and phrase structure representation (Bhatt et al., 2009) (Note that the dependency structure is manually annotated while the phrase structure is derived from the dependency trees). Multi-layered implies that there are different layers of representation: both syntax and lexical predicate argument structure are represented in this Treebank (Bhatt et al., 2011).

The manual annotation in the treebank is done using a dependency grammar framework. Rather than adopt a constituency based representation, where the sentence is first split into a noun phrase and a verb phrase, the verb is represented as the head of the syntactic tree. Arguments as well as adjuncts are its dependents. Languages like Hindi as well as Czech have adopted the dependency approach (Mel'čuk, 1988; Bharati et al., 1995) to build Treebanks.

The Hindi Dependency Treebank (HDT) uses an annotation scheme based on Computational

Paninian Grammar (CPG) (Begum et al., 2008). The Treebank consists of 400,000 words and is primarily newswire text, with a small amount of conversational data taken from works of fiction. The annotation represents syntactic dependencies in the form of modifier-modified relations. Most often, these consist of a verbal head with arguments as its dependents. For example, Figure 2.1 shows *Atif* and *kiṯaab* 'book' as dependents of the verb *paRii* 'read.perf.f.sg'.

It is noteworthy that the dependency labels depict relations between chunks, which are "minimal phrases consisting of correlated, inseparable entities" (Bharati et al., 2006). They are not necessarily individual words. In Figure 2.1, relations are depicted between a nominal chunk *Atif=ne* 'Atif=Erg' , with the verb *paRii* 'read.perf'. Verbal chunks typically consist of the lexical verb and its auxiliaries. The annotation of chunks also assumes that intra-chunk dependencies can be extracted automatically (Husain et al., 2010) e.g. the head of the nominal chunk *Atif* in *Atif=ne*.

Besides dependency relations, the Hindi Dependency Treebank 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)).

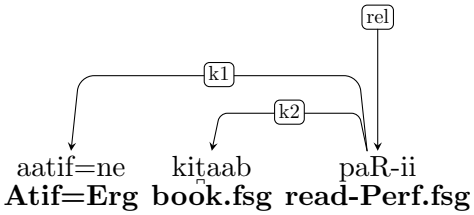

Figure 2.1: Dependency tree for *Atif=ne kiṯaab paRii* 'Atif read a book'. The head is *paRhii* 'read', with dependents `k1` 'karta' and `k2` 'karma'

The labels `k1` and `k2` in Figure 2.1 are also known as *karaka* labels. The term *karaka* is defined as "the role played by a participant in an action". The dependency relation labels have six *karaka* labels that represent core syntactic relations such as *karta* or `k1` 'locus of action' *karma* 'thing acted on', `k2`. Non-karaka labels include `rh` 'reason' or `k7t` 'time'.[1] Most *karaka* labels can only be understood in the context of a particular verb. For instance *Atif* and *kitaab* in Figure

[1] Although `k7t` is prefixed by 'k', it is not a *karaka* label

2.1 are considered participants in the event of reading. On the other hand, non-karaka labels are 'global', i.e. their interpretation will usually not vary across predicates. These usually correspond to adjuncts in the sentence. The tagset for dependency annotation also includes labels that indicate 'modifier' relations. For example, the relation between the nominal head and its modifying relative clause is represented by the label `nmod--relc`. This label is interpreted as a noun modifier relation of the type 'relative clause'.

Finally, the tagset also includes labels for non-dependencies. These represent other types of relations between two nodes in the dependency tree. The label `pof`, which is used for NVCs is one such relation. The `pof` label indicates a `p`art `of` unit. It shows that the pre-verbal element in a complex predicate forms a unit with its verbal head. In all, the dependency tagset consists of about 43 labels (Bharati et al., 2006).

There are nearly 3000 unique NVCs in the treebanked corpus, which have been manually annotated using the `pof` (**p**art-**of**) label. The NVC consists of two predicating structures, but the nominal is represented as a dependent of the light verb. However, the nominal can have its own arguments. Thus, the argument span for complex predicates includes not only direct dependents of the verb but also dependents of the noun (if any)(See also section (2.1.1)).

For example, in Figure 2.2 we have a complex predicate *pratikshaa kar* 'waiting do' ; 'wait'. The predicating noun *pratikshaa* 'waiting' has its own argument *Ravi kii*, indicated with the label `r6-k2`. The treebank has two labels `r6-k1` and `r6-k2` that are specifically used for the arguments of the predicating noun. However, in the Paninian framework, only those arguments of the nominal that have a genitive post position are represented as its dependents (Kulkarni, 2011). For example, *Ravi* in Figure 2.2 is followed by the genitive *kii* and is therefore represented as a dependent of *pratikshaa*. Nominal arguments that occur with any other postposition are always shown as dependents of the main verb.

The Hindi Treebank has annotated nouns, adjectives, borrowed words from English as well as compounds and adverbs with the `pof` label. This has resulted in a large amount of annotated data for NVCs. There are about 3015 unique combinations of these preverbal elements that combine

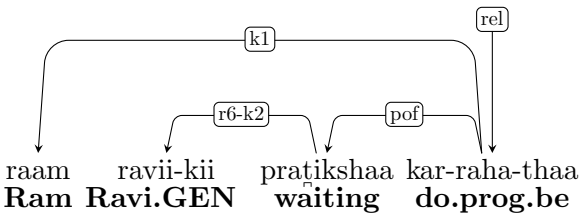

Figure 2.2: Dependency tree for NVC *pratikshaa* 'waiting' *kar* 'do'. The nominal predicate is linked to the light verb with a `pof` label

with a light verb in the Hindi Treebank. As more than one light verb can occur with the same pre-verbal host, we get 1884 cases if we count the pre-verbal host alone. Using part-of-speech information, we find that the majority of the pre-verbal hosts are nouns while others are adjectives and borrowed words from English.

POS	**Tokens**	**Types**
Noun	10, 425	1567
Adjective	5810	663
Borrowing (verbs)	143	65
Adverb	44	16

Table 2.1: Part-of-speech tags that occur with the `pof` label in the Hindi Treebank

In order to use the annotated Hindi Treebank data effectively, we had to filter out those NVCs cases that were not relevant for our analyses. Therefore, even though the `pof` was marked, we only took into consideration those constituents where the relation was marked between a head noun and verb. We ignored adjective- light verb or adverb-light verb cases. However, we did accept phrases containing borrowed English words, which are marked with the part of speech tag 'BLK' as they can form new NVCs in Hindi. In addition, we used the Hindi PropBank annotations as a quality control on the Hindi Treebank *pof* annotation.

2.1.1 Underspecification of NVCs

For NVCs, apart from the use of a label `pof`, the Treebank does not distinguish nouns in an NVC in any special way. The only place where there is a structural difference in the dependency

tree is when the nominal has an argument with the genitive case. If we look at example 10, the argument contributed by the nominal *Mohan* is marked with instrumental case. In 11, *Mohan* is marked with genitive.

(10) raam=ne mohan=se nafrat̲ k-ii
Ram.M.Sg=Erg Mohan.M.Sg-Inst hatred.F do-Perf.F.S
'Ram hated Mohan '

(11) raam=ne mohan=kii prashansaa k-ii
Ram.M.Sg=Erg Mohan.M-Sg=Gen praise.F do-Perf.F.S
'Ram praised Mohan '

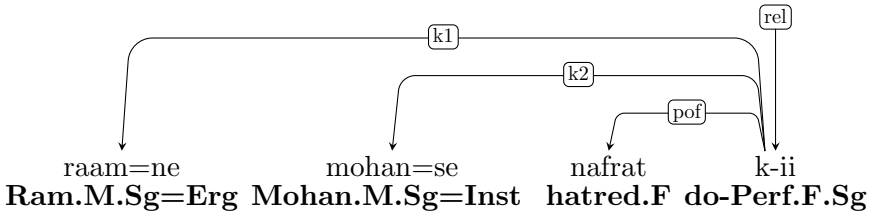

Figure 2.3: Dependency tree for NVC *nafrat* 'hatred' *kar* 'do'. The nominal's argument is a dependent of the light verb.

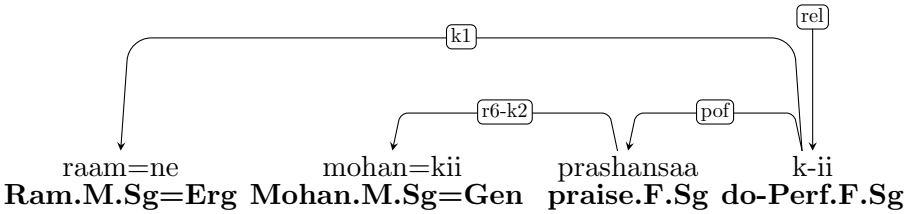

Figure 2.4: Dependency tree for NVC *prashansaa* 'praise' *kar* 'do'. The nominal's argument is a dependent of the nominal.

According to Computational Paninian Grammar (CPG) (Begum et al., 2008; Kulkarni, 2011), nouns sanctioning genitive arguments are considered true *verbal nouns*. Therefore their arguments are structurally dependent on the noun (Figure 2.4). If a nominal has non-genitive arguments, they are depicted as dependents of the verb (Figure 2.3). While this distinction is useful for capturing nominals that take genitive-marked arguments, the presence or absence of the genitive case alone

may not distinguish true nominal predicates from others. First, the genitive postposition is ambiguous with respect to arguments or adjuncts. For example, while a noun like *prashansaa* requires a genitive marked argument, others such as c^harc^haa 'debate' will take it optionally. Second, there appears to be no reason why a noun like *nafrat* should not be considered a true predicate because it takes an instrumental-marked argument rather than genitive. The postposition on the nominal's argument is a result of the lexical property of the nominal itself (Mohanan, 1997). We need other evidence to motivate a structural difference in the dependency representation of the NVCs in Hindi.

(12) mantrii=ne patrakaaron=se (c^{h}unaav=kii) c^{h}arc^{h}aa k-ii
minister.M.Sg=Erg reporter.M.Pl=Inst election.M.Sg=Gen discussion.F.Sg do-Perf.F.Sg
The minister had a discussion with the reporters (regarding the election)

2.2 Hindi PropBank

The Hindi PropBank (HPB) is concerned with the labeling of semantic roles. Semantic role labelling is done on top of the dependency trees in the HDT (Hindi Dependency Treebank). This task implies that the arguments for every predicate (including complex predicates) in the treebank must be assigned semantic roles. Semantic roles are defined on a verb-by-verb basis and description at the verb-specific level is fine-grained; e.g. a verb like *hit* will have 'hitter' and 'hittee'. These verb-specific roles are then grouped into broader categories using numbered arguments (`ARG#`). Each verb can also have a set of modifiers not specific to the verb (`ARGM*`). The distinction between numbered and non-numbered is quite similar to the broad division between *karaka* and non-*karaka* labels in the HDT. The verbs in the PropBank are annotated with predicate-argument structures and provide semantic role labels for each syntactic argument of a verb. Although these were deliberately chosen to be generic and theory-neutral (e.g., `ARG0`, `ARG1`), they are intended to consistently annotate the same semantic role across syntactic variations. For example, in both the sentences *John broke the window* and *The window broke*, *window* is annotated as `ARG1` and as bearing the role of Patient. This reflects the fact that this argument bears the same semantic role

in both cases, even though it is realized as the structural subject in one sentence and as the object in the other. In the Pāṇinian approach, the argument *window* in *The window broke* gets the same label as *John*. This is the primary difference between PropBank's approach to semantic role labels and the Pāṇinian approach to kāraka labels, which it otherwise resembles closely.

Later in this chapter, we also explore how this similarity may be used to expedite PropBank annotation in Section 2.3. Figure 2.5 shows the PropBank labels annotated on top of the dependency tree.

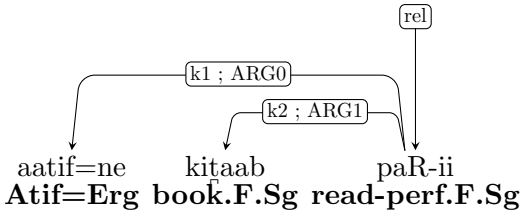

Figure 2.5: PropBank annotation for *paR* 'read', with dependents `ARG0` and `ARG1`

The numbered arguments in PropBank correspond to `ARG0-ARG3`, including function tags associated with `ARG2` (e.g. `ARG2-GOL`). `ARG0` and `ARG1`. 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. The objective is to capture semantic similarities across syntactic variation. If we look at the example *aatif-ne kitaab paṛii* 'Atif read a book' from the previous section, but this time annotated with semantic roles, we get the multi-layered annotation found in Figure 2.5.

The Hindi PropBank currently consists of 24 labels including both numbered arguments and modifiers (Table 2.2). The Hindi PropBank labels make certain distinctions that are not made in some other language such as English, e.g. `ARGA` and `ARGA-MNS` mark the arguments of morphological causatives in Hindi. The externalmost causer argument in case of multiple causers or the only causer argument present in the sentence is annotated as `ARGA`. Any intermediate causers are annotated with the `ArgA_MNS` label (Bhatia et al., 2013). The preverb in complex predicate constructions is annotated as `ARGM-VLV` for the verb-verb complex predicate and `ARGM-PRX` for the noun-verb

complex predicates. The ARGM-PRX label is similar across other PropBanks in Chinese and Arabic (Hwang et al., 2010).

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
ARG0-GOL	causee as recipient		
ARG0-MNS	causee		
ARGA-MNS	secondary causer		
ARGA	causer		
ARGM-VLV	verb-verb construction		
ARGM-PRX	noun-verb construction		
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 2.2: Hindi PropBank labels.

The annotation process for Hindi PropBank 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. In Table 2.3 PropBank-style semantic roles are listed for the simple verb *de*;'to give'. In the table, the numbered arguments correspond to the 'giver', 'thing given' and 'recipient'. Frame file definitions are created manually and include role information as well as a unique roleset ID (de.01), which is assigned to every sense of a verb. In addition, for Hindi the frame file also includes the transitive and causative forms of the verb (if any). Thus, the frame file for *de* will include *dilvaa* 'cause to give'. Although the forms are grouped under the same lemma form (e.g. *de*), in practice each transitive or causative variant will have its own unique roleset id. In English PropBank for instance, verb-particle constructions like *open up* are grouped under the same frameset *open*, but have unique roleset ids.

de.01	*to give*
`Arg0`	the giver
`Arg1`	thing given
`Arg2`	recipient

Table 2.3: A frame file

2.2.1 Manual validation of treebank NVC annotation

One of the challenges for PropBank annotation was validation of the `pof` label that was added to identify NVCs in the Hindi Treebank. As PropBank annotation involves frame file creation as a first step, the validation of NVCs was carried out at this stage. The process of frame file creation is described in chapter 3. Frame files were checked for annotation errors, such as invalid light verbs or inclusion of collocations. Certain borderline cases were further investigated using linguistic diagnostic tests. The application of linguistic tests revealed that NVCs do not pass the diagnostic tests in the same fashion.

We surveyed about 50 NVCs using seven linguistic diagnostic tests (We describe these in the sections below) and scored them on the basis of the number of tests they were able to pass. Table 2.4 shows these examples and the degree of variation with respect to their strength as a prototypical NVC. If we examine this table, we find that some NVCs appear to form a constituent that disallows syntactic modification of any kind. These are in the topmost row with a score of 7, for example *saamnaa* 'confrontation'. Other NVCs appear to fall somewhere along a cline that is neither a totally prototypical NVC case, nor an ordinary predicate argument structure.

A similar analysis was carried out for the NVCs with *ho* 'be', using only five linguistic diagnostic tests (as two of these can only be used for transitive NVCs). A similar pattern emerged, where some NVC cases were more prototypical than others (Table 2.5).

This shows that it is difficult to narrow down a definitive set of diagnostic tests that are the effective at identifying NVCs. The reasons for this are not yet clear, but our initial study showed that this could be related to the frequency of the NVC cases. In the NVCs with *kar* 'do', the 'low-scoring' i.e. less prototypical NVCs were more frequent in the corpus as opposed to the

NVC strength	Nominals with light verb *kar* 'do'
7	*saamnaa* 'confront', *apiil* 'appeal', *pra̲da̲n* 'confer', *d̲auraa*; 'tour', *ma̲da̲d̲* 'help', *samarthan* 'concur'
6	*pus^{h}ti* 'confirm', *pra̲da̲rs^{h}an* display, *prayas* 'effort', *hamla* 'attack', *j̃ac^h* 'investigate'
5	*mulaqaa̲t̲* 'meeting', *faislaa* 'decision', *kos^{h}is^h* 'effort', *g^{h}oshaṇaa* 'declaration', *baa̲t̲chiit̲* 'chat', *c^{h}arc^{h}aa* 'discuss', *baa̲t̲* 'talk', *virod^h* 'objection', *s^{h}uru* 'begin'
4	*d̲aavaa* 'tour', *maang* 'demand', *sviikaar* 'accept'
3	*vic^{h}aar* 'think', *inkaar* 'refusal'

Table 2.4: NVC scored according to the number of diagnostic tests that are cleared. These form predicates with *kar* 'do'. The highest score is 7 and lowest 3.

NVC strength	Nominals with light verb *ho* 'be'
5	*gat^{h}an* 'group', *nid^{h}an* death, *aayojan* 'organization'
4	*mau̲t̲* 'death', *bait^{h}ak* 'gathering', *pus^{h}ti* 'implementation', *gawahi* 'testimony', *baaris^h* 'rain'
3	*s^{h}uru* 'begin', *s^{h}uruaa̲t̲* 'beginning', *chori* 'theft', *prasaaraṇ* 'broadcast', *anuvaa̲d̲* 'translation', *ab^{h}yas* 'lesson', *amal* 'implement'
2	*c^{h}arc^{h}a* 'discussion', *nuksaan* 'destruction', *asar* 'effect', *zaruura̲t̲* 'need', *faayda̲* 'advantage'
1	*vrudd^{h}i* 'increase', *sunvaai* 'hearing', *anub^{h}uu̲t̲i* 'perception'

Table 2.5: NVCs with *ho* 'be' scored according to the number of diagnostic tests that are cleared. The highest score is 5 and the lowest is 0. Tests for transitive NVCs do not apply to *ho*

'high-scoring' NVCs (e.g. with a score greater than 5). Perhaps the NVCs lower on the scale are less tightly bound with the light verb and are ambiguous between an true NVC and an ordinary verb-argument usage in the language.

In the following subsections, we survey some of the diagnostic tests described by Mohanan (1994), Bhattacharyya et al. (2007) and Ahmed et al. (2012). We used these tests in our survey of NVC diagnostics described above. The 'extra argument' test was described in the previous chapter, therefore we list only six other diagnostic tests below. These tests show that the noun and light verb combinations form a complex predicate rather than an ordinary predicate argument structure. This implies that the noun and light verb are part of a constituent and the nominal does not have all the properties of an ordinary NP.

2.2.1.1 Apnaa-Apnaa test

The pronoun *apnaa* 'self' is a reflexive possessive that can be used with nominals that are not part of a complex predicate. If *apnaa-apnaa* can be added to the nominal, then it is not an NVC but an ordinary noun. In the example (13), *apnaa-apnaa* is added to the NVC *chorii* 'theft' *hu-ii* 'be-Perf.F.Sg'. As it is infelicitous, the noun and verb combination form a true NVC. Unlike the extra argument test (described in Chapter 1, section 1.5), this test can be used for intransitive NVCs.

(13) *apni apni corii hu-ii
self self theft.F.Sg be-Perf.F.Sg
'* Our own theft took place'

In example (13), the nominal 'chorii'; theft in the complex predicate 'chorii honaa'; theft happen; cannot have apnaa-apnaa as its modifier.

2.2.1.2 Wh-question test

If the noun forms a lexical category (rather than a regular NP), then a wh-word may not be substituted for it directly. Wh-interrogatives are formed with the wh-word in situ in Hindi. The

simple predicate example in (14) can be made interrogative by adding the form *kisne* 'who-Erg' in Example 15.

(14) mohan=ne ninaa=ko kiṯaab ḏ-ii
MohanM.Sg=Erg Nina.F.Sg=Acc book.F.Sg give-Perf.F.Sg
'Mohan gave the book to Nina'

(15) kis=ne Ninaa=ko kiṯaab ḏ-ii?
who-Erg Nina=Acc book.F.Sg give-Perf.F.Sg
'Who gave the book to Nina?'

In the case of complex predicates, the nominal hosts cannot be replaced by a Wh-question word like *kyaa* 'what' like example 17, where 'kyaa' replaces *bharosaa* 'trust' in the complex predicate *bharosaa kar* 'trust do; to trust (someone)'. Therefore, (17) cannot be an acceptable interrogative version of (16).

(16) ravi=ne mohan=par bharosaa ki-yaa
Ravi.M.Sg=Erg Mohan.M.Sg=loc reliance.M.Sg do-Perf.M.Sg
'Ravi relied on Mohan'

(17) *ravi=ne mohan=par **kyaa** ki-yaa?
Ravi.M.Sg=Erg Mohan.M.Sg=loc what do-Perf.M.Sg
'*What did Ravi do on Mohan?'

2.2.1.3 The nominal host cannot be relativized

Example 18 shows the application of the relativization test, proposed by Mohanan (1997). When the nominal is modified by the relative clause headed by *jo* the sentence is ungrammatical. This means that *yaad kii* is a true NVC because *yaad* cannot be relativized independently of the light verb.

(18) *vah yaad jo mohan=ki raam=ne kii ...
that memory that Mohan.M.Sg=Gen Ram.M.Sg=Erg story.F.Sg that .F.Sg do-perf.F.Sg
it
'(*)The story of Mohan that Ram remembered'

2.2.1.4 Addition of accusative case

The accusative marker test was proposed in Bhattacharyya et al. (2007) and it can be used with transitive light verbs such as *kar* 'do'. It is applied by adding the accusative marker to the nominal in the NV complex predicate. The accusative case, normally seen on the object noun should not appear on a nominal which is part of the complex predicate. If it does, one should assume that the NV does not form a complex predicate. In example 19, the nominal 'bharosaa' (in boldface) cannot get the accusative marker 'ko'. This example is ungrammatical.

(19) *usne raam par us **bharose ko** kiyaa jo..
he.erg Raam loc that trust acc do when.. '
'(Intended meaning) He placed that trust in Raam when..'

2.2.1.5 No Demonstrative modifier on nominal

If the nominal in the complex predicate is modified by a demonstrative like *yaha*; this, then it is an ordinary NVC rather than a complex predicate.

(20) raam=ne b^{h}arosaa **yaha** ki-yaa kii..
Ram.M.Sg=Erg trust.M.Sg this do-Perf.M.Sg that
'(*)Ram did this trust that.. '

In the example 20, the nominal *b^{h}arosaa* cannot be modified by the demonstrative *yaha*.

2.2.1.6 Adjectival modification of the nominal

Mohanan (1997) describes the adjectival modification test, where predicating nominals cannot be modified by adjectives such as *ek* 'one'. This test is not a very strong diagnostic as we found many predicating nouns that would allow modification by an adjective.

(21) *mohan=ne raam=ko **ek** k^{h}abar ki-ii
Mohan.M.Sg=Erg Ram.M.Sg=Dat one news.F.Sg do-Perf.F.Sg
'*Mohan gave a news to Ram'

In Example 21, *k^{h}abar* cannot be modified by the numeral *ek*.

2.3 *Karaka* Labels and PropBank Mapping

PropBank annotation has been carried out in Chinese, Arabic and Korean besides English. For all of these languages, PropBank annotation is done on Penn Treebank style **phrase** structure (Xue and Palmer, 2003; Palmer et al., 2008). On the other hand, for Hindi, annotation is carried out on top of **dependency** trees. This representation has some advantages for a task such as PropBank. First, arguments of the verb are marked explicitly as dependents, and this makes the task of argument identification easier as compared to phrase structure. Second, the Hindi Treebank (HDT) provides a rich set of dependency relation labels. This facilitates mappings between syntactic dependents and semantic arguments. 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).

As mentioned earlier, both numbered arguments and *karaka* labels are meaningful in the context of a particular verb, i.e. these are verb specific labels. In this respect, numbered arguments `ARG0-ARG3` and *karaka* labels `k1-7` are quite similar but there is an important exception. For example, HDT treats the following sentences similarly, whereas PropBank does not:

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

The boy and *the window* are both considered `k1` for HDT, whereas HPB 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, unaccusative verbs are not treated differently. 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 Hindi PropBank (HPB) does not make such a distinction, `k4a` maps to `ARG0`.[2] The `ARG0` and `ARG1` mapping would be accurate only when we make use of specific verb

[2]We recently revised the analysis for Experiencer subjects and now have ARG0-GOL instead of ARG0. This was not implemented when the mapping experiments were carried out

information. We reproduce in Table 2.6 the mapping between dependency labels and PropBank found in Vaidya et al. (2011). In contrast to the numbered argument mapping which is complex for ARG0, the non-numbered arguments are relatively straightforward. The mapping for non-numbered or 'modifier' arguments may be found in Table 2.7.

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

Table 2.6: Numbered argument mappings. Note that the ARGC label mapping represented an earlier analysis of causatives in Hindi PropBank. This label is no longer part of the current tagset, but it is retained here as it formed the basis of our experiments in Vaidya et al. (2011).

For the remaining arguments i.e. non-numbered arguments and non-karaka labels, we get a one-to-one mapping based on the definitions of these labels alone. We do not have a verb-specific resource for these cases as these represent the adjuncts in the sentence. In contrast to the verb specific or local semantic roles like the numbered arguments, non-numbered arguments are global semantic roles (Vaidya and Husain, 2011). Despite this one-to-one correspondence, in practice these labels are far harder to identify automatically. This is because in annotation practice the interpretation of definitions like 'cause' or 'purpose' tends to differ, hence the Treebank data for these labels can be quite noisy.

2.3.1 Mapping Rules: Experiments

This section describes the result of experiments that implemented the dependency to PropBank mapping using a small corpus of treebanked and PropBanked data (Vaidya et al., 2011). The goal was to determine the extent to which the mapping could expedite the process of PropBank annotation. Our corpus consisted of simple predicates as well as complex predicates.

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

Table 2.7: Modifier mappings.

As a first step, we carried out argument identification (i.e. identifying candidate phrases for annotation) followed by argument classification (selecting the correct PropBank label for the argument). Argument identification was easy, given the dependency treebank. For each verbal predicate, we considered all syntactic dependents of the predicate as its semantic arguments. Argument identification was done with high precision as both PropBank and HDT follow the same principles for identifying syntactic dependents and semantic role labels for a verb. Argument classification was done by using two types of rules:

- Empirically-derived rules, generated by measuring frequency of dependency features in association with semantic roles in a small annotated corpus.
- Linguistic-mapping rules, derived from our linguistic intuitions described earlier (section 2.3)

2.3.1.1 Empirically-derived rules

The empirically derived rules relied on a corpus of annotated data. We used a subset of the Hindi Dependency Treebank that was used in the ICON'10 contest (Husain et al., 2010). Our corpus contained about 32,300 word tokens and 2,005 verb predicates, in which 546 were complex predicates. Each verb predicate was annotated with a verb sense specified in its corresponding frameset file. Note that at this stage, we had annotated complex predicates without nominal frame file information. This annotation was only for experimental purposes, therefore we annotated all

the arguments of the light verb, with the nominal getting the default label `ARGM-PRX`. While this does not reflect our procedure for NVC annotation (which does take into account the nominal's frame file), it does demonstrate that the mapping can be applied to both types of predicates. At that stage, there were only 160 frameset files created for the verbal predicates. All verbal predicates (including complex predicates) were annotated with PropBank labels. A total of 5,375 arguments were annotated.

This corpus formed the basis for extracting various features. Three kinds of features were used for the generation of empirically-derived rules: predicate ID, predicate's voice type, and argument's dependency label. Predicate ID is either the lemma or the roleset ID of the predicate. Predicate lemmas are already provided in the Treebank. When we used roleset IDs, we assumed that sense annotations were already done. PropBank includes annotations of verb sense, called roleset ID, that differentiates 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 IDs, we generated more fine-grained rules that are specific to those senses.

A predicate's voice type is either 'active' or 'passive', also provided in the Treebank. There were not many instances of passive construction in our current data, which made it difficult to generate rules general enough for future data. However, even with the small amount of training instances, we found some advantage to using the voice feature in our experiments. Finally, an argument's dependency label is the dependency label of an identified argument with respect to its predicate. These rules were 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 (*run*, 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 $rule(run, \text{active}, \texttt{k1}) = \texttt{ARG0}$.

Since we do not want to apply rules with low confidence, we set a threshold to $P(pbrel)$ so that predictions with low probabilities can be filtered out. Finding the right threshold is a task of handling the precision/recall trade-off. [3] For our experiments, we ran 10-fold cross-validation to find the best threshold for our case. In general, the higher the threshold, the higher and lower the precision and recall become, respectively. Figure 2.6 shows comparisons between precision and recall with respect to different thresholds. Notice that the threshold of 1.0, meaning that using only rules with 100% confidence, does not give the highest precision. We decided to stick with a threshold of 0.93.

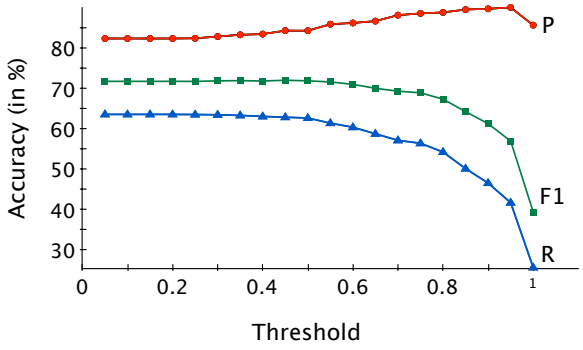

Figure 2.6: Accuracies with respect to different thresholds. P, R, and F1 stand for precisions, recalls, and F1-scores.

2.3.1.2 Linguistic Mapping rules

Mapping rules are applied to arguments that the empirically-derived rules cannot classify. These rules capture general correlations between syntactic and semantic arguments for each pred-

[3] We calculate precision by using the following formula, where TP stands for true positives and FP stands for false positives: $Precision = TP/TP + FP$. Precision tells us about the proportion of cases that were correctly identified. Recall is calculated by finding out the total number of true positives and dividing them by the total number of elements that actually belong to the positive class i.e. $Recall = TP/TP + FN$, where FN = false negatives. The F1 score is calculated by the harmonic mean of Precision and Recall. $F1 = 2 * (Precision * Recall)/(Precision + Recall)$

Roleset	Usage	Rule
A.01	to come	k1 → ARG1 k2p → ARG2-GOL
A.03	to arrive	k1 → ARG1 k2p → ARG2-GOL k5 → ARG2-SOU

Table 2.8: Rules for the predicate 'A (*come*)'.

icate and are helpful for predicates not seen in training data. The numbered argument mappings are directly entered into the PropBank frameset files manually by the frameset file creators. This serves as a useful resource for our mapping experiments, but the files include mappings for numbered arguments only. Table 2.8 shows mappings for the predicate 'A (*come*)', specified in the frameset file, 'A-v.xml'. For example, the predicate 'A' has two 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.

2.3.2 Evaluation

Since there is a relatively small size of data, we do not make a separate set for evaluation. Instead, we run 10-fold cross-validation to evaluate our rule-based system.

Table 2.9 shows accuracies achieved by the system. We achieve the best result when the linguistic mapping rules from section 2.3.1.2 are applied after the empirically derived rules (section 2.3.1.1). The mapping rules improve the recall of numbered arguments (ARGN) significantly, but result in a slight decrease in their precision.

	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 2.9: Labeling accuracies achieved by both rules. We set our threshold at 0.93. 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 by only the empirically derived rules.

The precision and recall results for ARG0 and ARG1, are better than expected, despite the complexity of the mapping. This is because they occur most often in the corpus, so enough rules are extracted to be useful. The other numbered arguments are more closely related to particular classes of verbs (e.g. motion verbs for ARG2-GOL, LOC). The mapping for all numbered arguments improves overall with the addition of linguistically motivated rules. We would expect the modifiers to map independently of the verb, but experiments showed that the presence of the verb enhanced overall performance. Although Table 2.7 expects a one-to-one mapping of modifiers (according to definition), it does not happen in practice.

We observe that the coarse or fine grained interpretation of labels plays an important role in both frameworks. For example, in PropBank the mapping rule for ARGM-ADV performs poorly because it is used for a variety of sentential modifiers, including conditional clauses and it can map to as many as four HDT labels. On the other hand, PropBank makes a distinction between means and causes using ARGM-CAU and ARGM-MNS. HDT chooses not to make such distinctions.

Our comparison of the HDT labels with PropBank is beneficial for the annotation process for Hindi. Accurate mapping can speed up annotation and can possibly make the labelling more consistent and accurate. At the same time, the mapping to semantic roles will be useful for other tasks such as the creation of frame files for non-verbal predicates.

2.4 Other Resources

Apart from the Hindi Treebank and PropBank, we also utilized other corpora and semantic resources for Hindi. We utilized mainly web-based corpora as the size of the Hindi Treebank is relatively small.

2.4.1 Web-based corpora

The Hindi Wikipedia, [4] was used to create a corpus consisting of 10 million words. This corpus was downloaded and processed using the WikiExtractor [5]and the LXML libraries in Python. The second was the BBC Hindi corpus, scraped from web, consisting of about 7 million words. We POS-tagged this corpus using the freely available tagger described in Reddy and Sharoff (2011).

In order to compute statistical features such as log-likelihood, we made use of a larger 60 million word Hindi web corpus, distributed under the Sketch Engine license (Kilgarriff et al., 2010). This corpus is includes Hindi Wikipedia and web text and was also POS-tagged.

2.4.2 Hindi WordNet

Hindi WordNet consists of 82,000 words organized in 33900 synsets (Bhattacharyya, 2010). We utilize the Hindi WordNet in order to look up semantic relations for the nominals that occur as part of NVCs. Semantic relations such as hypernymy and hyponymy can be useful to understand the semantic category of a nominal. The Hindi WordNet also specifies approximately 170 ontological categories for each synset. These consist of a tuple such as *Abstract, Inanimate,Noun* or *Cognition, Abstract, Inanimate, Noun*. The ontological category describes a property in descending order of generality. In most of our experiments we utilized the first property specified in the tuple e.g. *Abstract* or *Cognition*.

[4]http://dumps.wikimedia.org/hiwiki

[5]http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

Chapter 3

Lexical Resource Representation

3.1 Lexical Resources

While there are only around 700 unique simple verbs in Hindi, there are many more complex predicates in the language. Representing these cases in a lexical resource can be challenging, not only because of their high frequency, but also because new NVCs can be coined easily and quickly. NVC representation in resources like dictionaries, valency lexicons, WordNets and computational grammars requires an understanding of their lexical and syntactic behaviour. To some extent, this has been addressed for NVCs (and other multi-words) in English (Sag et al., 2002; Copestake et al., 2002). For Hindi as well, there has been some recent work on the representation of NVCs (Hwang et al., 2010; Ahmed et al., 2012). The number of NVCs in Hindi far exceed those in English, therefore the lexical representation challenge is at a larger scale.

3.2 Representing NVCs in PropBank

The annotation of NVCs in corpora implies that every instance of an NVC needs to be manually identified and labelled. In the Hindi Treebank, the identification of NVCs has already been carried out. The Hindi PropBank (HPB) is concerned with the labeling of semantic roles. Semantic role labelling is done on top of the dependency trees in the Hindi Dependency Treebank. This task implies that the arguments for every predicate (including complex predicates) in the treebank must be assigned semantic roles. Semantic roles are defined on a verb-by-verb basis and description at the verb-specific level is fine-grained (For more details on Hindi PropBank and

Treebank, including the tagset, see Chapter 2).

An NVC such as *pratikshaa kar* 'waiting do; wait' in the Treebank is already identified with the `pof` label as shown in Figure 3.1. Its arguments will be annotated with PropBank labels as shown in Figure 3.1.

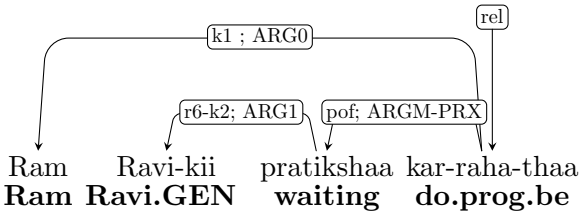

Figure 3.1: Dependency tree for NVC *pratikshaa* 'waiting' *kar* 'do'. The nominal predicate is linked to the light verb with a `pof` label. PropBank annotation is shown on top of the dependency labels.

Before this annotation is carried out, PropBank subcategorization frames or 'frame files' need to be defined for each instance of an NVC. PropBank annotation creates frame files for the nominals, which also contain information about the light verb that combines with them. For PropBanking in particular, similar lexical resources for nominal predicates have been created for English and Arabic (Hwang et al., 2010). When building the frame files, we more or less followed a 'listing' approach, where every NVC that appeared in the treebank was defined with an individual roleset in a frame. This provided full coverage, along with semantic role information for each instance of the NVC in the Treebank. The following section describes the design of the nominal frame files to create this lexical resource.

3.3 Nominal frame files

In order to create the nominal frame files for Hindi PropBank, we considered every occurrence of a nominal in an NVC with a given light verb. We can compare example (22) with example (23) below:

(22) raam=ne cycle=kii chorii k-ii
Ram.M.Sg=Erg cycle.F.Sg=Gen theft do-Perf.F.Sg
'Ram stole a bicycle'

(23) aaj cycle=kii chorii hu-ii
Today cycle.F.Sg=Gen theft be.Part-Perf.F.Sg
'Today a bicycle was stolen'

In example 22 we get an agentive subject with the light verb *kar* 'do'. However, when it is replaced by the unaccusative *ho* 'become' in Example 23, then the resulting clause has a theme argument as its subject. Note that the nominal *chorii* in both examples remains the same. From the point of view of PropBank annotation, the NVC *chorii kii* will have both `ARG0` and `ARG1`, but *chorii huii* will only have `ARG1` for its single argument *cycle*. Hence, the frame file for a given nominal must make reference to the type of light verb that occurs with it. As both noun and light verb contribute to the semantic roles of their arguments, we require linguistic knowledge about both parts of the NVC. The semantic roles for the nominal need to specify the co-occurring light verb and the nominal's argument roles must also be captured. Table 3.1 describes the desired representation for a nominal frame file.

Frame file for *chorii*-n(oun)	
chorii.01: theft-n	light verb: *kar*'do; to steal'
`Arg0` `Arg1`	person who steals thing stolen
chorii.02 : theft-n	light verb: *ho* 'be/become; to get stolen'
`Arg1`	thing stolen

Table 3.1: Frame file for predicate noun chorii 'theft' with two frequently occurring light verbs *ho* and *kar*. If other light verbs are found to occur, they are added as additional rolesets as chorii.03, chorii.04 and so on.

This frame file shows the representation of a nominal *chorii* 'theft' that can occur in combination with a light verb *kar* 'do' or *ho* 'happen'. For each combination, we derive a different set of PropBank roles: agent and patient for chorii.01 and theme for chorii.02. Note that the nominal's frame actually contains the roles for the combination of nominal and light verb, and not the nominal alone.

Nominal frame files such as these have already been defined for English PropBank.[1] However, for English, many nominals in NVCs are in fact nominalizations of full verbs, which makes it far easier to derive their frame files (e.g. *walk* in *take a walk* is a noun derived from a full verb). In fact, in the most current representation of PropBank frame files for English, verbs and their nominalizations are unified into a single frame (Bonial et al., 2014).

For Hindi, most nouns in NVCs are not nominalizations, hence a correspondence with the verbal frames is not an obvious solution. In Hindi, as NVCs are highly productive, with nearly 3000 unique types of NVCs in a 400,000 word corpus, the process of manual frame file creation can be very time consuming. As frame file creation is a pre-requisite for annotation, we need to have a different strategy for defining semantic roles for NVCs- one that reduces manual effort and also incorporates linguistic knowledge from other existing resources.

Therefore, as a first step, we decided to partially automate the process of frame file creation. A linguistic mapping between the dependency *karaka* labels in the Treebank and PropBank labels is the basis for deriving semantic roles for nominals in Hindi. The automatically generated semantic roles will be used to create frame files for annotation. This method reduces the effort required for manual creation of these files and accurately predicts semantic roles for almost two thirds of the unique nominal-verb combinations, with around 20% partial predictions, giving us a total of 87% useful predictions. For the implementation, we use linguistic resources in the form of syntactic dependency labels from the treebank as well as existing frame files for **simple** verbs[2] in Hindi (Vaidya et al., 2013).

3.3.1 Generating semantic roles

This section describes a novel method for generating PropBank semantic roles for each complex predicate in the corpus. The Hindi Treebank has already identified NVC cases by using a special label `pof` or 'part-of'. We use the label given by the Treebank as a means of extracting the

[1] http://verbs.colorado.edu/propbank/framesets-noun/

[2] http://verbs.colorado.edu/propbank/framesets-hindi/

NVC cases. Once this extraction step is complete, we have a set of nominals and a corresponding list of light verbs that occur with them.

We use two resources to derive linguistic knowledge about the NVC: PropBank frame files for simple verbs in Hindi and the Hindi Treebank, annotated with dependency labels.

3.3.1.1 Mapping from the Hindi Treebank

The Hindi Treebank is used as a source of argument structure information for the NVCs. We first extract all the dependency *karaka* label combinations that occur with a unique instance of an NVC. For example, the NVC *chorii karnaa* 'theft do' ; steal can occur with a set of karaka labels such as `k1` karta (actor), `k2` (locus of action), `k7t` temporal modifier. Of these, only `k1` and `k2` are argument labels, whereas `k7t` is an adjunct. Therefore, `k7t` gets filtered out. Every instance of *chorii karnaa* is used to calculate the most frequently occurring combination of labels that occur with it. As a result, we get a tuple consisting of *chorii karnaa*, a set of *karaka* argument labels that occur with it and a count of the number of times that NVC has occurred in the corpus. E.g. ("chorii karnaa", k1:k2, 12)

The *karaka* labels are then mapped onto PropBank labels following the procedure described in chapter 2.

3.3.1.2 Mapping from Verb frames

Our second resource consists of PropBank frames for full Hindi verbs. Every light verb that occurs in Hindi is also used as a full verb, e.g. *de* 'give' may be used both as a 'full' verb as well as a 'light' verb. As a full verb, it has a frame file in Hindi PropBank. The set of roles in the full verb frame is used to generate a "canonical" verb frame for the each light verb. The argument structure of the light verb will change when combined with a nominal, which contributes its own arguments. However, as a default, the canonical argument structure list captures the fact that most *kar* 'do' light verbs are likely to occur with the roles `ARG0` and `ARG1` respectively or that *ho*'become', an unaccusative verb, occurs with only `ARG1`.

3.3.1.3 Procedure

Our procedure integrates the two resources described above. First, the tuple consisting of *karaka* labels for a particular NVC is mapped to PropBank labels. But many NVC cases occur just once in the corpus and the *karaka* label tuple may not be very reliable. Hence, the likelihood that the mapped tuple accurately depicts the correct semantic frame is not very high. Secondly, Hindi can drop arguments in a sentence e.g., *(vo) kitaab paRegaa*; '(He) will read the book' (see also section 1.2). These are not inserted by the dependency annotation (Bhatia et al., 2010) and are not easy to discover automatically (Vaidya et al., 2012). We cannot afford to ignore any of the low frequency cases as each NVC in the corpus must be annotated with semantic roles. In order to get reasonable predictions for each NVC, we use a simple rule. We carry out a mapping from karaka to PropBank labels only if the NVC occurs at least 30 times in the corpus. If the NVC occurs fewer than 30 times, then we use the "canonical" verb list.

3.4 Manual correction of NVC annotations

The automatic method described in the previous section generated 1942 nominal frame files. In order to evaluate their accuracy, we opted for manual checking of the automatically generated frames. The frame files were checked by three linguists and the checking focused on the validity of the semantic roles. The linguists also indicated whether annotation errors or duplicates were present. While the Hindi Treebank made use of the Extra argument diagnostic to annotate the `pof` label, there were several annotations that were ordinary noun-verb combinations. This was either due to the fact that the verb in the NVC did not appear to be a true light verb e.g. *kaha* 'say or *bigar* 'spoil'. Or, the noun and light verb were collocations rather than NVCs e.g. *chaapa maara* 'stamp hit; to stamp'. For such borderline cases, the linguists applied some of the diagnostics described in chapter 2, section 2.2.1. If they were not NVCs, these were marked as 'Errors' during frame file validation.

3.4.1 Evaluation

The automatic method described in the previous section generated 1942 nominal frame files. After manual checking of these cases (as described in chapter 2), the total number of frame files stood at 1884. These frame files consisted of 3015 rolesets i.e. individual combinations of a nominal with a light verb (see Table 3.1). The original automatically generated rolesets were compared with their hand corrected counterparts (i.e. manually checked 'gold' rolesets) and evaluated for accuracy. We used three parameters to compare the gold rolesets with the automatically generated ones: a full match, partial match and no match. Table 3.2 shows the results derived from each resource (Section 3) and the total accuracy.

Type of Match	**Full**	**Partial**	**None**	**Errors**
Karaka Mapping	25	31	4	0
Verbal Frames	1929	642	249	143
Totals	1954	673	245	143
% Overall	65	22	8	5

Table 3.2: Automatic mapping results, total frames=3015

The results show that almost two thirds of the semantic roles are guessed correctly by the automatic method, with an additional 22% partial predictions, giving us a total of 87% useful predictions. Only 8% show no match at all between the automatically generated labels and the gold labels. The 'Errors' column indicates cases that are not NVCs and have been filtered out during manual validation.

When we compare the contribution of the karaka labels with the verb frames, we find that the verb frames contribute to the majority of the full matches. The karaka mapping contributes relatively less as only 62 NVC types occur more than 30 times in the corpus. If we reduce our frequency requirement from of 30 to 5, the accuracy drops by 5%. The bulk of the cases are thus derived from the simple verb frames. We think that the detailed information in the verb frames, such as unaccusativity, contributes towards generating the correct frame files.

It is interesting to observe that nearly 65% accuracy can be achieved from the verbal information alone. The treebank has two light verbs that occur with high frequency i.e. *kar* 'do' and

Light verb	Full (%)	None (%)	Total Uses*
kar'do'	**64**	8	1038
ho 'be/become'	**81**	3	549
de 'give'	55	**34**	157
A 'come'	31	**42**	36

Table 3.3: Light verbs 'do' and 'be/become' vs. 'give' and 'come'. *The unique total light verb usages in the corpus

ho 'become'. These combine with a variety of nominals but perform more consistently than light verbs such as *de* 'give' or *A* 'come'. The light verb *kar* adds intentionality to the NVC, but appears less often with a set of semantic roles that are quite different from its original 'full' verb usage. In comparison, the light verbs such as *de* 'give' show far more variation, and as seen from Table 3.3, will match with automatically derived frames to a lesser extent. The set of nominals that occur in combination with *kar* usually seem to require only a doer and a thing done.

3.5 Discussion

The outcome of the NVC semantic role generation is the availability of gold standard frame files for around 3000 NVCs in Hindi. In addition, our frame file generation process acts as a quality control on the `pof` annotation of the Hindi Treebank. The frame files are being used for semantic role annotation. During annotation, a given instance of an NVC is annotated by referring to the semantic roles listed in the frame file. While annotation is greatly helped by the presence of the frame files, there are still several practical difficulties during the annotation process.

As both noun and light verb must be taken into consideration, the process of argument identification (i.e. distinguishing between arguments and adjuncts) is more challenging. Consequently, argument labelling as well becomes more difficult. While on one hand, the frame file supplies semantic roles, it would be more helpful to have semantic information at a more abstract level e.g. with respect to membership with a class or group of nouns. This will make the process of argument identification more consistent- as we would expect a particular group of nouns to project similar semantic roles. Subsequently, this would also be more helpful for training automatic semantic role

labellers. As there are many individual instances of NVCs, learning semantic roles associated with a single NVC will encounter the problem of data sparsity. But using information about noun groups might make this process less dependent on individual training instances.

This is also one of the main questions with respect to lexical representation i.e. a choice between a simple listing approach vs. a more generalizable and predictive approach. On one hand, the simple listing of NVCs is sometimes unavoidable (as seen in Section 3.2), but this also suffers from a "lexical proliferation problem" (Sag et al., 2002). For example, frequently-occurring NVCs would usually be captured in a list, but this list would not take into account generalizable linguistic properties. In the following chapter, we make an exploratory attempt to discover noun groups or templates for NVCs.

Chapter 4

Finding Predicative Noun Groups

4.1 Introduction

We discussed the creation of nominal frame files in the previous chapter. While the creation of individual frames for every noun and light verb combination is useful for annotation, in practice it is still a difficult task to distinguish between arguments and adjuncts. Therefore, it is useful to have as a reference some over-arching generalizations that predict similar semantic roles and arguments for a given group of nouns. In other words, a notion of noun classes for predicative nouns would help in generating more consistent annotations.

Levin (1993)'s classic work on verb classes is based on the idea that "particular syntactic properties are associated with verbs of a certain semantic type" Levin (1993)[5]. English verb classes that have been developed on the basis of this idea have been converted and extended into VerbNet (Kipper et al., 2008), which has been widely used as a computational resource for tasks such as semantic role labelling (Swier and Stevenson, 2004) to natural language generation (Habash et al., 2003). English FrameNet (Baker et al., 1998) is a resource that defines frames or similar conceptual structures for lexical items, such as predicative nouns. For Hindi, such resources are unavailable and moreover, developing them would require manual effort and linguistic expertise.

In this chapter, we will examine whether it might be possible to arrive at some semantic patterns for Hindi nouns using a simple clustering algorithm that uses linguistic features. While our results are still preliminary, we have made use of the resulting noun classes as a feature for identifying NVCs in corpora (see Chapter 6). At the end of this chapter, we describe our strategy

for using noun classes to expand the lexicon in a computational grammar (Sulger and Vaidya, 2014).

4.2 Previous Work

Linguistic generalizations about Hindi NVCs have not been studied very well, but some recent studies have addressed this problem. Ahmed and Butt (2011) use manual methods and linguistic introspection and Butt et al. (2012) have used a combination of manual and statistical methods.

Ahmed and Butt (2011) have suggested that the combinatory possibilities of N-V combinations are in part governed by the lexical semantic compatibility of the noun with the verb. Similar observations have been made for English (Barrett and Davis, 2003; North, 2005). Ahmed and Butt (2011) look at the light verbs *kar* 'do', *ho* 'be', *hu-* 'become' and identify three classes of nouns based on co-occurrence patters. The first consists of psychological nouns such as *yaad* 'memory' that occur with all three light verbs. The examples shown in 24-26 represent the class that is compatible with all of the light verbs surveyed.

(24) nadya=ne kahaani yaad k-ii
Nadya.F.Sg=Erg story.F.Sg memory.F.Sg do-Perf.F.Sg
'Nadya remembered a/the story (agentively).' (lit. 'Nadya did memory of a/the story.')

(25) nadya=ko kahaani yaad hɛ
Nadya.F.Sg-Dat story.F.Sg memory.F.Sg be.Pres.3.Sg
'Nadya remembers/knows a/the story.' (lit. 'At Nadya is memory of a/the story.')

(26) nadya=ko kahaani yaad hu-ii
Nadya.F.Sg-Dat story.F.Sg memory.F.Sg be.Part-Perf.F.Sg
'Nadya came to remember a/the story.' (lit. 'The memory of a/the story came to Nadya.')

Other NVC classes may only be compatible with a subset of light verbs. The second and third classes consist of nouns that are classified as more or less agentive in nature- based on their capacity to form CPs with *hu-* 'become'. For example, the noun *tamir* 'construction' is only compatible with the light verb *kar* 'do' but disallows *hu-* 'become' (examples 27 and 28).

(27) bilaal=ne makaan taamir ki-yaa
Bilal.M.Sg-Erg house.M.Sg construction.F.Sg do-Perf.M.Sg
'Bilal built a/the house.'

(28) *bilaal=ko makaan taamir hɛ/hu-aa
Bilal.M.Sg-Dat house.M.Sg construction.F.Sg be.Pres.3.Sg/be.Part-Perf.M.Sg
'Bilal built a/the house.'

This work clearly shows that compatibility with a light verb can be an important feature to identify an NVC class. Based on the combinatorial possibility and acceptability with the light verb, it is possible to identify groups of nouns with similar properties.

In a follow-up paper, Butt et al. (2012) attempted to identify these classes of Urdu N-V CPs automatically. However, one of the drawbacks of their method was the use of an untagged corpus, which required extensive filtering in order to separate the light and non-light instances of these verbs. After filtering out the irrelevant combinations, by making use of a visualization tool, they found that most nouns were either psychological nouns (and occurred with all three light verbs) or nouns that were highly agentive and disallowed *hu-* 'become'. This study took into consideration the three light verbs from the Ahmed and Butt (2011) study.

A recent study on Persian NVCs (which are similar in many ways to Hindi) uses distributional vector-space methods to find similar nouns (Taslimipoor et al., 2012). It showed that verb vectors are a very useful indicator of noun similarity. The reported results are significantly better using the light verb dimension; Taslimipoor et al. (2012) state that this affirms their original intuition that a verb-based vector space model can better capture similarities across NVC.

We will draw upon the results of the papers described above to motivate this present work. The classes identified by Ahmed and Butt (2011) seem promising, but the corpus work was done manually, and the total size of their data set is limited to 45 nouns. In constructing the noun classes, we take a different route in that we try to expand the search space by using an external,

manually-crafted resource (i.e. the Hindi Treebank) and a larger set of light verbs to come up with more substantial noun groups.

4.3 Noun Clustering

In our experiment, the aim is to use a clustering algorithm, specifically, k-means to explore similarities between groups of nouns. We limit our study to those nouns that are annotated as NVCs in the Treebank. This is advantageous because we know that we are only looking at those noun and verb combinations that are NVCs.

4.3.1 Data

The Hindi portion of the Hindi and Urdu Treebank (Bhatt et al., 2009) includes NVCs that have been manually tagged with the dependency label POF (which stands for "part of"). The POF label is used for adjectives and adverbs as well as nouns. We extracted POF cases that were nouns only. We extracted an initial list of around 504 high frequency nouns after removing spelling variations and annotation errors. For our study, we chose a list of frequently occurring light verbs (shown below). We then used the treebank to extract the co-occurrence patterns of each noun against the light verbs in this list and calculated their relative frequencies as attributes for each noun.

(29) *ho* 'be', *kar* 'do', *de* 'give', *le* 'take', *rakh* 'put', *lag* 'attach', *a* 'come'

In addition, we also looked for the ontological node description for each of these nouns in Hindi WordNet (Bhattacharyya, 2010). For example, the noun *varsha* 'rain' has the ontological node description as 'Natural_State,State,Noun'. WordNet contains around 170 unique ontological node descriptions for every synset. If a noun belongs to a particular synset, we chose the topmost node as our attribute e.g. for *varsha* 'rain', this would be 'Natural_State'. For some nouns, this ontological information was not present, so the initial list of 504 nouns was reduced to 464 nouns. In future work, the ontological nodes can be added manually for the missing nouns. In all, the 464 nouns were

associated with 46 unique ontological categories. A complete list of the 46 ontological categories can be found in appendix A.

4.3.2 Gold classes

We would like to compare the automatically found noun groups to a set of gold classes that describe an ideal partitioning of this data. We found that it was not an easy task to determine these gold classes and we explored several possibilities. First, we utilized WordNet relations such as co-hyponymy and hypernymy, following a study on Dutch nouns (Van de Cruys, 2006). We used these relations to create gold classes for nouns, but this resulted in many false classes. Next, we translated Hindi nouns to English, and then mapped them to English VerbNet classes via the English unified frames (Bonial et al., 2014). But this method also failed, due to the difficulty in translating nouns into English. Finally, we opted to create gold standard classes by making use of the manually validated noun frames that were described in chapter 3. If nouns project similar semantic roles, then we might surmise that they are also semantically similar in some way. The manually validated noun frames are a resource that can be used as a basis for comparing the automatically discovered noun groups. For example if Noun1 and Noun2 both contain the roles `ARG0` and `ARG1` in their frame file, then we can say that these nouns are likely to be similar.

Note that PropBank labels are generally quite coarse-grained and they usually mark prototypical agents and patients. More detailed roles viz. Agent, Patient, Theme etc. might give us narrower semantic classes. In addition, the frames themselves are limited in that they do not contain every possible semantic role combination that can be found for these nouns. However, unless one is able to handcraft a resource with precisely such a goal, the frames are the next best alternative.

Therefore, for each of the 464 nouns in our training data, we prepared gold classes using PropBank semantic frames. Table 4.1 shows these classes derived from the frame files, the number of nouns that occurred in each class and an example.

Each noun class consists of a label set e.g. `A0A1` corresponds to the fact that this group of

Class Label	Explanation	Count	Example
A0A1	Occurs with Arg0, Arg1	141	*prashansaa* 'praise'
A0A1A2	Occurs with Arg0,Arg1, Arg2	27	*puraskaar* 'award'
A1_A0A1	Occurs with Arg1 and Arg0,Arg1	119	*virodha* 'protest'
A0A1_A0A1A2	Occurs with Arg0,Arg1 and Arg0,Arg1,Arg2	48	*prastaava* 'proposition'
A0	Occurs only with Arg0	11	*snaana* 'bath'
A0A2LOCSOU	Occurs with Arg0 and Arg2-Location or Arg2-Source	17	*rishvata* 'bribe'
A1	Occurs only with Arg1	39	*maut* 'death'
A1_A0A1_A0A1A2	Occurs with Arg1 and Arg0,Arg1 and Arg0,Arg1,Arg2	34	*vishvaas* 'trust'
Misc	Occurs with a mixed number and type of label	19	*bachata* 'savings'
A1_A0A1A2	Occurs with Arg1 and Arg0,Arg1,Arg2	9	*parichay* 'introduction'

Table 4.1: Gold classes for 464 nouns extracted from PropBank Noun frames

nouns occurs with the labels `Arg0` and `Arg1` in their frameset. A class label like `A0A1_A0A1A2` shows that nouns occur with two label sets viz `Arg0` and `Arg1` as well as `Arg0`,`Arg1` and `Arg2`. In this way, we found 9 clear classes of labels using the framesets and one 'Miscellaneous' class, which contained nouns that did not correspond to any one particular set of labels. Most of the nouns in the study occurred in framesets with the label set `A0A1`. Label sets such as these were also used for deriving a verb classification for the Chinese Treebank (Xue and Kulick, 2003).

4.3.3 Noun Clustering using Weka

Clustering algorithms are used to partition data into groups or clusters. Manning and Schütze (1999) describe two uses for these algorithms viz. exploratory data analysis and forming generalizations. For the Hindi nouns, we are interested in both these objectives. For this experiment we chose to use the k-means clustering algorithm (MacQueen, 1967) as it is usually the first to be used on a data set that is relatively unexplored.

The k-means algorithm works at partitioning the observations in the training set into k clusters, so as to minimize the within cluster sum of squared errors (WCSS). When k-means is implemented, it usually consists of an assignment step, where each observation in the data is assigned to a particular cluster. This assignment is determined by whether the cluster's centroid and the observation have the lowest within cluster sum of squared error. Post assignment, the next step is to calculate new centroids and each compare each observation to the new centroids. These

steps are iterated until every observation is as close as possible to its centroid.

K-means is quite sensitive to its initial random assignment of observations to clusters. The random assignments also imply that centroids of these clusters will not be the same each time the algorithm is run. This is why, it is necessary to run k-means many times (i.e. over a large number of iterations). Based on the actual implementation of k-means, we may also require the first cluster's centroids or 'seeds' to be specified and often the results are also quite sensitive to this seed value.

In comparison to other clustering algorithms, k-means carries out hard clustering. This implies that a given observation can belong to only one cluster at any given time. This has some disadvantages for linguistic data that may sometimes belong to multiple groups.

In our experiment we made use of the Weka implementation of k-means (Hall et al., 2009). While the initialization process is still random, we make use of the k-means++ initialization parameter provided by the tool. This ensured that initialization is based on specific probabilities for the initial centers and reduces the chance of getting very different results every time the algorithm is run. Other parameters such as the size of k need to be specified beforehand for the k-means algorithm. The size of k denotes the number of required clusters/partitions in the data, which must be supplied beforehand.

Simple k-means tries to minimize the sum of squared errors in order to converge. However, there are variations of k-means, which can use measures other than the sum of squared errors. Schulte im Walde and Brew (2002)'s experiments utilized distance measures such as skew divergence and information radius, in a variation of k-means. Such measures may be more suited to language data and can be used for further experiments with Hindi nouns.

In our training file, each noun is assigned an ID as well as a list of attributes viz. the relative frequencies of the seven light verbs described earlier and the ontological description. The ontological category is a nominal attribute whereas the relative frequencies are numerical attributes. When Weka is given a categorial (or nominal) attribute like the ontological category, it will consider the distance between two nominal values as zero if they are exactly the same, and 1 if they differ. We experimented with using binary features instead of the categorial feature for the ontological

category, but they did not result in any improvement in performance.

4.3.4 Cluster Evaluation

We ran the k-means implementation of Weka several times over our data using different values for k, specifically, $k = 2 - 15$. In order to identify the optimum value of k, we used the total of the within cluster sum of squared errors (WCSS) which is provided in the Weka output. As we get closer to the optimum value of k, the compactness of the automatically found clusters increases, causing a sudden drop in the WCSS for each cluster. In contrast, those values that are larger than the optimum value of k will not result any substantial change in WCSS (although there will be a gradual decrease). This gradual decrease indicates that the clusters are not getting any more compact. If we plot the WCSS against the size of k, then we can look for a sudden drop in WCSS, followed by a gradual decrease. That point is likely to be the ideal value of k for the data.

Using WCSS, we were able to identify the optimum value of k=4 for the training data with two features viz. the ontological property and relative frequencies of light verb (Figure 4.1). Using relative frequency only, we found the best k value as 5 (Figure 4.2).

Once we have identified the optimum value of k, we also calculated the precision, recall and F-measure for these clusters. This was done by creating a contingency table, which tabulates the number of nouns found in each cluster against their gold classes. In Table 4.2, we show the performance for the ontological feature and relative frequency together and relative frequency by itself. The F1 score of the combined feature is slightly better than the relative frequency feature alone. The ontological feature helps in the slight improvement of recall as compared to the precision.

	Size of k	**Precision**	**Recall**	**F-measure**
Onto+RelFreq	4	36.51	54.54	**43.82**
RelFreq	5	36.32	49.47	41.88

Table 4.2: Precision, Recall and F-measure for each feature group

If we look at the cluster-specific performance for each of these features, we find that regardless of the features, there are three classes that k-means can identify more accurately. For other classes,

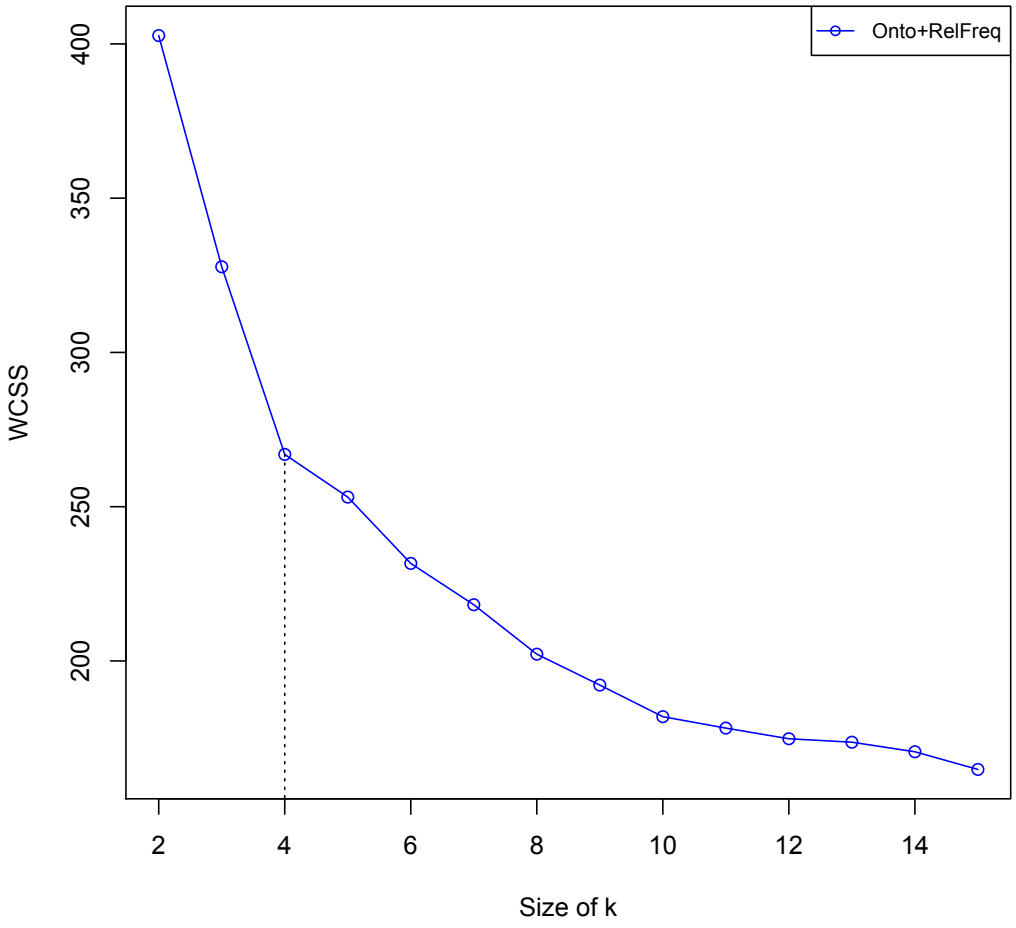

Figure 4.1: Within cluster sum of squared errors plotted against the size of K. The 'knee' at $k = 4$ shows that the cluster size of 4 is likely to be the ideal size for this data.

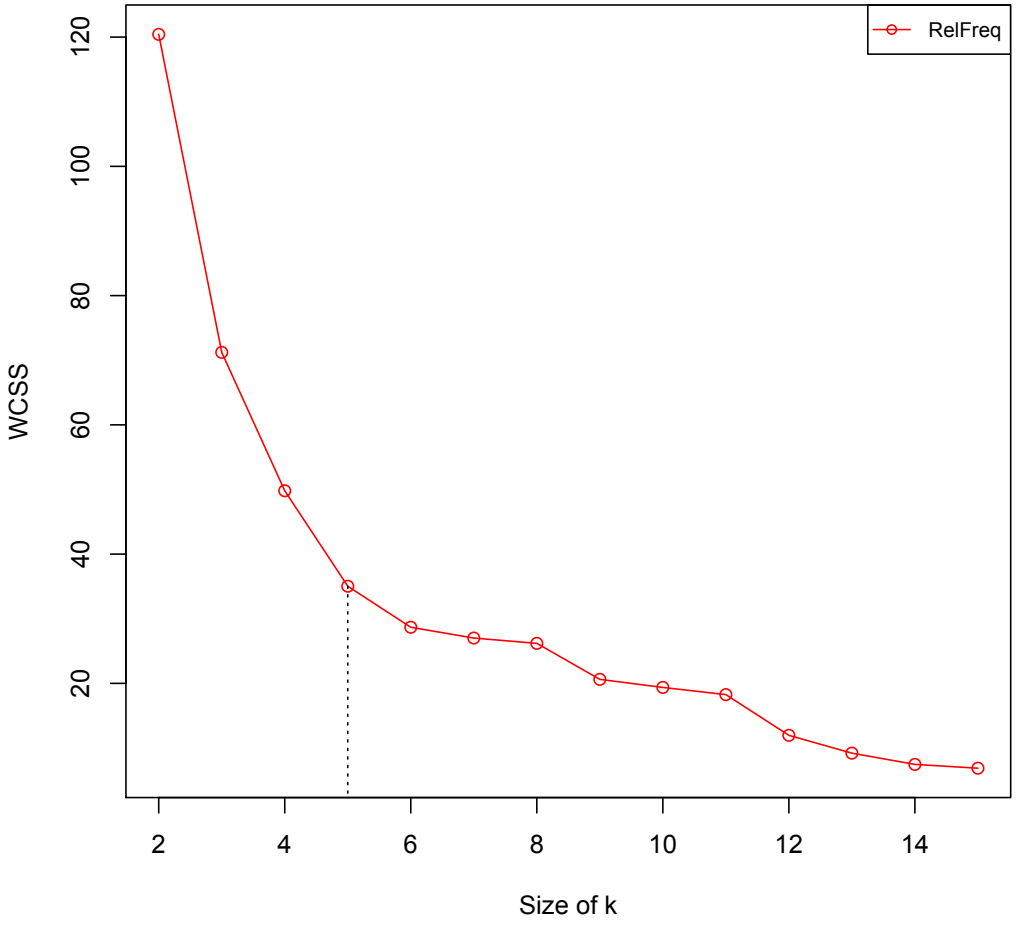

Figure 4.2: Within cluster sum of squared errors for the relative frequency feature plotted against the size of K. The 'knee' at $k = 5$ shows that the cluster size of 5 is likely to be the ideal size for this data.

it finds that these features are not helpful or give very poor results. This is because light verbs *ho* 'be', *kar* 'do' and *de* 'give' can occur with a very large number of nouns in the data.

Table 4.3 shows that there were the three classes `A1`, `AOA1` and `AOA1_A1` that were identified accurately. For each of these three classes, the light verb centroids are *ho* 'be', *kar* 'do' and *kar* and *ho* together.

If we look more closely at the clusters for these classes, we find that they contain nouns with somewhat more distinctive properties. For instance, class `A1` consists of nouns that have the ontological property 'Cognition' or 'Mental_State'. The class `AOA1` describes nouns with the property 'Physical, Action,Abstract,Inanimate' and in `AOA1_A1`, we find nouns with the property 'Action,Abstract,Inanimate'. The former property is a subtype of the latter, showing that the `AOA1` class consists of nouns that are more eventive than `AOA1_A1`.

Gold class	**Onto+RelFreq**			**Properties (Centroids)**
	Precision	Recall	F1	
A1	32.97	79.48	46.6	State, ho 'be'
A0A1	46.10	50.35	48.13	Physical, kar 'do'
A0A1_A1	41.07	47.89	44.21	Action, kar 'do', ho 'be'
A0A1_A0A1A2	25.97	41.66	31.99	Communication, de 'give'

Table 4.3: Class-to-cluster evaluation for the combined features- ontology and relative frequency (k=4)

The `AOA1_A1` class has the light verbs *ho* and *kar* as its prominent centroids. For example, nouns like *sahayoga* 'help' have the property 'Action, Abstract, Inanimate' and can alternate frequently between either *kar* 'do' or *ho* 'be'. Thus, the same noun occurs with either light verb to result in a change in the number of arguments. It is interesting to observe that this alternation is captured by the eventive property of the nouns. We discuss these nouns in more detail in Chapter 5.

At the same time, this eventivity is differentiated from the class `AOA1` alone, where nouns like *khoja* 'discovery' appear with the label 'Physical, Action, Abstract, Inanimate'. These are nouns that describe an event that is more highly agentive in nature. On the other hand, nouns such as

pareshaani 'discomposure' appear with the label 'Cognition, Abstract, Inanimate' in the class `A1`. These nouns are primarily stative.

For the remaining cluster, the performance is not as good. The `A0A1_A0A1A2` cluster has as its centroid the light verb *de* 'give'. This cluster contains several nouns with the ontological property of 'Communication' e.g. nouns such as *vachana* 'promise' or *dhamakii* 'threat'. These are the prototypical examples that describe a communicating event. The fact that *de* 'give' appears with more than one label set points to the fact that the light verb *de* 'give' is quite variable cross-linguistically, e.g. in English, it is possible to find a number of NVCs with varying semantic roles e.g. *give a sigh*, *give a speech* or *give an explanation*. Nouns occurring with *de* 'give' show a greater variation in their semantic roles and are harder to cluster more accurately (See also, Table 3.3 in chapter 3). The noun clusters nevertheless give us an indication of the salient semantic properties of each group. The k-means algorithm is able to replicate the linguistic generalizations found in (Ahmed and Butt, 2011) over a larger set of predicating nominals. This confirms that eventivity vs. stativity and agentivity vs. experience of the action are determiners of semantic classes among nouns.

4.4 Noun Groups in Hindi/Urdu Grammar Development

In this section we demonstrate how the nominal classes described in the previous section can be utilized in a computational grammar. We look at the Hindi/Urdu ParGram Grammar, which is part of a larger international research effort, the ParGram (Parallel Grammars) project (Butt et al., 1999, 2002; Butt and King, 2007). All of the grammars in the ParGram project are couched within the Lexical Functional Grammar framework and are implemented using the development platform XLE (Crouch et al., 2012a). The grammars are developed manually and not via learning methods, which allows for a theoretically sound analysis that is also efficient from a computational point of view. The Hindi/Urdu ParGram Grammar aims at covering both Hindi and Urdu, which is a design decision that suggests itself due to the many structural conformities of the two languages (Butt et al., 2002).

One weakness of the grammar is its currently relatively small lexicon, compared to other ParGram grammars. Adding to the lexicon is a critical step in extending the grammar coverage. This is even more true for NVCs due to the high frequency of such constructions in running text. Thus, we see the Hindi/Urdu ParGram Grammar as an ideal test bed for developing a lexical resource of Hindi nouns.

The implementation of a computational grammar for a language that makes heavy use of NVCs calls for two requirements. First, the lexical items taking part in NVC formation need to be present in the lexicon of the grammar; and second, the grammar needs to be engineered in a way that represents the correct linguistic generalizations. Any approach that is short of either of these requirements will result either in loss in coverage or overgeneration of the grammar. Therefore, knowledge about the combinatorial possibilities of NVCs will help in defining lexicon entries and templates in a more compact manner.

Our clustering experiments have shown that nouns appear with several different distributions, often with one dominant light verb, but also with the possibility of occurring with one or two other light verbs. The clusters do not represent absolute certainties about NVCs, but report **tendencies** of occurrences. For example, a predicating nominal from the list of 504 nouns may be assigned to the gold class `AOA1`. In that case, it is most likely to occur with light verbs *kar* 'do' (as this is the cluster centroid in Table 4.3). It is probably less likely to occur with any of the other light verbs in our study e.g. *ho* 'be' or *aa* 'come'.

We will discuss how such tendencies of occurrences can be integrated into the LFG grammar via the construction of templates. These templates can be augmented to model the relevant linguistic generalizations in terms of constraints inspired by optimality theory (OT, Prince and Smolensky (2004)). A serious evaluation of the effect on the grammar of adding in this lexical resource is planned for future work.

4.4.1 Templates in XLE

In XLE, grammar writers can define templates in a special section of the grammar that can be called from the lexicon. Templates allow generalizations to be captured and, if necessary, changes to be made only once, namely to the template itself (Butt et al., 1999; Dalrymple et al., 2004). Consider the template in 30, which models intransitive verbs in English; these are represented in LFG terms as predicates that apply to a single grammatical function, a subject. The lexical entry in 31 for the English intransitive verb *laugh* calls up the `INTRANS` template; the argument supplied to the template is substituted for the `P`(redicate) value inside the template definition.

(30)
```
INTRANS(P) = (^ PRED) = '_P<(^ SUBJ)>'
@NOPASS.
```

(31)
```
laugh V @(INTRANS laugh).
```

In ParGram grammars, the lexicons are generally organized so that each verb subcategorization frame corresponds to a different template. Similarly, templates can be defined to encode a given set of generalizations about how certain groups of nouns combine with different light verbs. Consider the NVCs in (32) and (33). The noun *istemaal* 'use' forms part of the cluster dominated by *kar* 'do' and thus occurs most frequently with *kar* 'do' as well as *ho* 'be'.

(32) nadya=ne c^{h}amac^{h}=kaa istemaal ki-yaa
Nadya.F.Sg=Erg spoon.M.Sg=Gen use.M.Sg do-Perf.M.Sg
'Nadya used a spoon'

(33) c^{h}amac^{h}=kaa istemaal hu-aa
spoon.M.Sg=Gen use.M.Sg be-Perf.M.Sg
'A spoon was used.'

The lexical entry of the noun *istemaal* 'use' is given in (34).[1] The entry points to the template `NVGROUP2` which is defined as in (35). This version of the template constrains the verbal type of

[1]The transliteration scheme employed in the Hindi/Urdu ParGram Grammar is described in Malik et al. (2010).

the overall predication to be a CP either with the light verb *do* 'give' or the light verb *ho* 'be', or to not be a CP at all.

(34)
```
istemAl NOUN-S XLE (^ PRED) = 'istemAl<(^ OBJ)>'
                        @NVGROUP2.
```

(35)
```
NVGROUP2 = { (^ VTYPE COMPLEX-PRED-FORM) =c kar
           |(^ VTYPE COMPLEX-PRED-FORM) =c ho
           | ~ (^ VTYPE COMPLEX-PRED-FORM)}
```

4.4.2 Preferred CPs

The template in (35), however, misses out on the fact that for all the groups identified, there are N-V combinations that are more productive (and thus more likely to be CP constructions) than other combinations (which are more likely to be non-CP constructions, e.g., plain objects). In XLE, grammar developers can model statistical generalizations using special marks that were inspired by Optimality Theory (Prince and Smolensky, 2004). On top of the classical constraint system of existing LFG grammars, (formulated in terms of feature constraints stated on the f-structure), a separate projection, o-structure, determines a preference ranking on the set of analyses for a given input sentence. A relative ranking is specified for the constraints that appear in the o-projection, and this ranking serves to determine the winner among the competing candidates. The constraints are also referred to as OT marks and are overlaid on the existing grammar (Frank et al., 1998).

OT marks can be added in the appropriate place in the grammar to punish or prefer a certain analysis. For example, (36) states that `Mark1` is a member of the optimality projection. The order of preference of a sequence of OT marks can be specified in the configuration section of the grammar; an example preference ordering is given in (37). Here, the list given in `OPTIMALITYORDER` shows the relative importance of the marks. In this case `Mark5` is the most important, and `Mark1` is the least important. Marks that have a `+` in front of them are preference marks. The more preference marks that an analysis has, the better. All other marks are dispreference marks (the fewer, the better).

(36) `... Mark1 $ o::* ...`

(37) `OPTIMALITYORDER Mark5 Mark4 Mark3 +Mark2 +Mark1.`

Given the relative ordering of light verb tendencies in our noun groups, we can augment the templates with OT marks that represent such tendencies. The noun template in (35) is changed in two ways. First, **all** the light verbs are included; second, each disjunct is extended by two OT marks that represent the statistical likelihood of this particular combination forming a CP or not.[2] The ordering of the marks is shown in (39), where the mark `cp-dispref` is most severely punished, and the mark `+cp-pref` is most strongly preferred. With an ordering like this, a CP analysis for (32) is preferred, while a compositional analysis is dispreferred by XLE; the inverse will apply to *istemaal lag*, which is not a CP.

(38)
```
NVGROUP2 = { { (^ VTYPE COMPLEX-PRED-FORM) =c kar
               cp-pref $ ::*
               | ~ (^ VTYPE COMPLEX-PRED-FORM)
               non-cp-dispref $ ::* }
               ...
               |(^ VTYPE COMPLEX-PRED-FORM) =c lag}.
               cp-dispref $ o::*
               | ~ (^ VTYPE COMPLEX-PRED-FORM)
               non-cp-pref $ ::* } }.
```

(39) `OPTIMALITYORDER cp-dispref non-cp-dispref +cp-pref +non-cp-pref.`

Note that since OT orderings represent statistical tendencies rather than absolute constraints, they cannot be used to rule out certain combinations as ungrammatical. Thus, while being dispreferred, the non-CP analysis for *istemaal lag* will still show up in XLE output. This is in fact

[2]For space reasons, only the disjuncts for *de* 'give' as well as *lag* 'attach' are shown.

a welcome feature, since CP formation seems to happen on a continuum of constructions — a theoretical assertion that requires more work, certainly.

4.5 Discussion

In this chapter, we have investigated the creation of noun classes for predicative nouns that participate in NVC formation. The primary motivation for this work arose out of a lack of linguistic generalizations for these nouns in Hindi. Our clustering experiments showed that linguistic features such as co-occurrence with a particular light verb can predict about 4 out of the 10 gold classes with some accuracy. While the clustering results do capture some broad tendencies with the noun groupings, they miss certain alternations as well. For example, the clustering results show that psychological or stative nouns appear more often with *ho* 'be' alone. However, it is also true that nouns such as *vichaar* 'thought' are mental state or psychological state nouns that can occur with both *kar* and *ho*. This pattern is the opposite of the one described above, but was not clear enough to be found automatically.

We also demonstrated the creation of noun templates for the Hindi/Urdu LFG grammar as a potential application of our clustering result. We take advantage of the fact that the lexicons in LFG are organized such that each verb subcategorization frame corresponds to a different template (e.g. transitive or intransitive). Similarly, templates can be defined to encode a given set of generalizations about how certain groups of nouns combine with different light verbs. In Sulger and Vaidya (2014) we described the design of a noun template consisting of an ordered list of light verbs. The template describes the likelihood of that noun forming a valid NVC with a particular light verb. If a given noun and light verb combination occurs more frequently in the corpus, we prefer an NVC analysis. If it occurs rarely, the grammar will parse the sentence as a simple noun object and verb combination.

In terms of future work, there are a number of possible directions. The inclusion of more morphosyntactic features such as the postpositions on the NVC's arguments may help in identifying classes more accurately. The semantic information from ontological categories in WordNet is

partially useful, but perhaps more abstract semantic information is required. Another important possibility is the construction of hand crafted noun classes for a smaller number of nouns (e.g. 50), which may be used as a benchmark. Then, the PropBank 'label sets' can themselves be used as features. We would also like to experiment with clustering algorithms other than k-means that might be able to take into account light verbs that occur with low frequency.

Chapter 5

Lexicalized Grammars and NVCs

5.1 Representation of NVCs

In the previous chapter, we described a method of integrating valid NVC combinations into a computational grammar viz. ParGram based on the Lexical Functional Grammar (LFG) framework. In this chapter,we focus instead on the syntactic representation of the NVCs themselves using lexicalized grammars. Although NVCs have been given a structural representation in the Hindi and Urdu Dependency Treebank, this is underspecified (See chapter 2, section 2.1). The Treebank uses the single label `pof` to distinguish predicating pre-verbal elements from ordinary arguments. Apart from this distinction, the nominal itself is not distinguished in any other way. While it may be possible to modify the dependency analysis to capture its syntactic properties fully, we chose to follow a different route.

We explore lexicalized grammars, which contain 'deep' or linguistically precise characterizations of phenomena like NVCs. Examples of such grammars are Lexical Functional Grammar (LFG), Tree-Adjoining Grammar (TAG) and Head driven Phrase Structure Grammar (HPSG). An additional attractive feature of these grammars is that they are computationally tractable and can be used to provide automatic parses for a given language. For Hindi, we are also interested in comparing the analyses given by using one or more of the lexicalized grammars. There are existing theoretical accounts of Hindi NVCs in the literature (Mohanan, 1997; Davison, 2005; Bhatt, 2008), but these have not been compared (either across languages or formalisms).

The Hindi NVC presents two challenges with respect to syntactic representation: first, both

noun and light verb are predicating elements and contribute arguments to the NVC, resulting in a complex argument structure. Second, the nominal element in the Hindi NVC can act as a predicating element as well as an argument of the light verb simultaneously.

Within the LFG framework itself, an analysis for the representation of NVCs has already been worked out (Ahmed et al., 2012). We will describe this approach and then compare it with an existing analysis for NVCs in Tree-Adjoining Grammar (TAG) (Han and Rambow, 2000). We then discuss which of these two analyses may be the most appropriate for the Hindi NVC data. Our analysis is parallel to two recent proposals to extract a lexicalized LFG and TAG grammar respectively from the Hindi Treebank (Hautli et al., 2012; Bhatt et al., 2012). While we do not implement such a grammar extraction, the present work can inform such a task in the future (See also (Xia, 1999)).

We will begin this chapter with a brief overview of the NVC data. The existing literature on NVCs points towards two separate approaches for NVC representation viz. noun-centric and verb-centric approaches. These depend on the importance given to either the noun or the verb component in an NVC. Following this, we describe an existing verb-centric approach in Lexical Functional Grammar (LFG) and compare it with a noun-centric analysis in Tree-Adjoining Grammar (TAG). We then compare the two approaches for Hindi and present our conclusions.

5.2 Data

Predicating nominals in NVCs can combine with a range of light verbs as seen in Chapter 4. In this chapter, however, we look at a subset of these nominals that can combine with only two light verbs. Specifically, we make use of the noun classifications presented in Ahmed and Butt (2011) and Mohanan (1997) to focus on a group of nominals that alternate with *kar* 'do' and *ho* 'be'.

5.2.1 Alternation with *kar* and *ho*

The light verbs *kar* 'do' and *ho* 'be' are the most commonly occurring light verbs in Hindi. They can combine with a range of nouns to form NVCs. In chapter 4, we examined the noun classification in Ahmed and Butt (2011). Their classification criteria consist of syntactic combinatorial possibilities with light verbs *kar* 'do', *hu-* 'become' and *hɛ* 'be' (The forms *hu-* 'become' and *hɛ* 'be' are the eventive and stative forms of the light verb *ho* 'be' respectively). Three noun classes are described in Ahmed and Butt (2011), based on the combinatorial criteria described above. The first class consists of psychological nouns like *shaq* 'suspicion', which combine with all three forms as shown in (40), (42) and (41).

(40) raam=ne mohan=par shaq ki-yaa
Ram.M.Sg=Erg Mohan.M.Sg=Loc distrust.m do-Perf.M.Sg
'Ram distrusted Mohan'

(41) ram=ko mohan=par shaq hu-aa
Ram.M.Sg=Dat Mohan.M.Sg=Loc distrust.M be.Part-Perf.M.Sg
'Ram came to be distrustful of Mohan/experienced distrust of Mohan'

(42) ram=ko mohan=par shaq hɛ
Ram.M.Sg=dat Mohan.M.Sg=loc distrust.m be.Pres.M.Sg
'Ram is distrustful of Mohan'

Yet another class of nouns (class 'B'), in (43) can combine with *kar* but these nouns do not alternate with a dative subject as seen in (44). Compare this with the psychological noun *shaq* in example (41). Class B nouns can combine with *hu-* but not *hɛ*. When they combin with *hu-*, they have a pre-supposed agentive subject (45). The example in (45) is a case where combination with *hu-* has an intransitivizing effect. Such nouns may also be found cross-linguistically. For example, in English, a similar alternation structure may be found in *bring to light* vs. *come to light* (Claridge, 2000). Here, two light verbs *bring* and *come* are used to express either a causative or inchoative reading. In Persian, *kardan* 'make or do' and *ŝodan* 'become' alternate with the same noun in a manner similar (although not identical) to Hindi. In this chapter, we focus mainly on nouns from

Class 'B' like *t̪areef* 'praise' that show this alternation.

(43) logon=ne pust̪ak=kii t̪areef k-ii
people.M=Erg book.F=Gen praise.F do-Perf.F
'People praised the book '

(44) *logon=ko pust̪ak=kii t̪areef hu-ii/hɛ
people.M=Dat book.F.Sg=Gen praise.F be.Part-Perf.F.Sg/be.Pres.3.Sg
'*People experienced the book's praise '

(45) pust̪ak=kii t̪areef hu-ii/*hɛ
book.F.Sg=Gen praise.F be.Part-Perf.F.Sg/be.Pres
'The book was praised'

The final third class of nouns (Class 'C') combines with *kar* 'do' and *hɛ* 'be' but not with *hu-* 'become'. Hindi also has nouns like *baarish* 'rain' that combine only with the light verb *hu-*, in (46) and not with *kar* 'do', but we do not focus on these nouns in this chapter.

(46) aaj raat baarish hu-ii
today night.F rain.F be.Part-Perf.F.Sg
'It rained tonight'

5.2.2 Agreement

In addition to classifying nouns based on combinatorial possibilties, Mohanan (1997) also points out another division among the predicating nominals in NVCs. She shows that a class of nominals such as *yaad* 'memory' or *istemaal* 'use' can optionally take non-subject arguments with nominative or accusative case. When this occurs, the light verb will not agree with the predicating nominal (see 47). When the same noun, however, takes non-subject arguments with genitive case in (48), the light verb agrees with the nominal.

(47) Light verb does not agree with nominal

sitaa=ne mohan=ko **yaad** ki-yaa.
Sitaa.F.Sg=Erg Mohan.M.Sg=Acc memory.F do-Perf.M.Sg

'Sita remembered Mohan '

(48) Light verb agrees with the nominal

sitaa=ne mohan=kii **yaad** k-ii.
Sitaa.F.Sg-Erg Mohan.M.Sg=Gen memory.F do-Perf.F.Sg

'Sita remembered Mohan '

Nouns showing this type of optional agreement also differ from nominals (such as *ṭareef* 'praise') with respect to sentential negation, gapping and passivization Mohanan (1997). (We refer the reader to the tests shown in Mohanan (1997), and do not repeat them here). This split among NVCs also points to another property of the syntactic behaviour of NVCs. The light verbs in (48) or (43) will agree with the nouns, which in turn act as predicating elements as well as arguments *simultaneously*. On the other hand, the noun *yaad* in (47) will not be an argument of the light verb.

Davison (2005) describes a list of around 20 nouns that behave like *yaad*. In the Hindi Treebank, we performed a search for all NVC cases with *kar*, which had an accusative-marked object argument. We found about 100 such nouns, which include examples such as *kshamaa* 'forgiveness' as well as several borrowed nouns such as *design*, *phone* and *blackmail*. The accusative marker *ko* is understood to be a marker of specificity or animacy when it occurs in a transitive verb with an ergative subject (Bhatt and Anagnostopoulou, 1996). Accordingly, the examples in the Treebank are also in a discourse context where the object argument refers to a specific thing or person.

From the point of view of syntactic description, we now need to consider two specific phenomena with respect to Hindi NVCs : first, the representation of argument structure as a result of alternation with two light verbs and second, the phenomenon of the light verb that can optionally

agree with the predicating nominal. With these facts in mind, we now look at an existing analysis of the NVC in the Lexical Functional Grammar framework.

5.3 LFG analysis of NVCs

The work of Mohanan (1994); Alsina et al. (1997) and Butt (1995) has focused on the joint predication of noun and light verb in an NVC, where both parts of the predication contribute their meaning. Butt (1993) views the light verb as a unique category that shares a lexical entry with its non-light form. It is resolved as light or non-light depending upon the syntactic environment. When a verb is resolved as light, it supplies additional aspectual or agentive information about the event, and the host provides additional eventive meanings. These syntactic and semantic interactions with the host change the event structure of the ordinary predicate into a complex predicate.

The status of the light verb differentiates the analysis in Lexical Functional Grammar (LFG) from analyses such as Grimshaw and Mester (1988); Kearns (1988). They assume that the light verb inherits arguments from the nominal and its only function is to supply verbal case to the semantic arguments of the nominal. In effect, the light verb has to give up all its predicating power. LFG considers the light verb as a non-empty element which contributes a specific lexical meaning to the NVC. For example, the light verb may signal forcefulness, volitionality or surprise (Hook, 1974). Butt (1993) amasses cross-linguistic evidence to show that a light verb forms a syntactic class that is quite separate from an auxiliary or simple predicate. Moreover, light verbs are diachronically stable, having co-existed with their 'full' verb counterparts for a long period of time (Butt and Lahiri, 2013).

Therefore, the LFG analysis considers the light verb and its pre-verbal element as co-predicators in a single clause (Butt, 1993, 1995). The mechanism that is used to capture the process of co-predication exploits the levels of representation that are available in LFG. Syntactic representation is factored apart into two (or more) levels of representation that will also interact with each other. Most commonly, these appear as f(unctional)-structure and c(onstituent)-structure. F-structure represents predicate argument relations in terms of grammatical functions such as sub-

ject and object. C-structure on the other hand models the linear order of words and hierarchical relations between constituents.

In addition, the f-structure can be related to an additional level (or projection) known as a-structure, which describes predicate-argument structure in terms of thematic roles (Bresnan and Zaenen, 1990). The a-structure has been used to model complex predicates in Urdu (Butt, 1995; Butt et al., 2008). Example 49, reproduced from Butt et al. (2008) shows an example of an NVC in Urdu, which has a borrowed English nominal *pinch* and a light verb *kar* 'do'.

(49) bacce=ne haathi pinch ki-yaa
child.M.Sg=Erg elephant.M.Sg pinch do-Perf.M.Sg
'The child pinched a/an elephant.'

Butt et al. (2008) describe the a-structures for the co-predicators *pinch* and *kar* in Figure 5.1. The %Pred shows that the argument structure of the light verb is incomplete. The nominal's argument structure will be substituted here resulting in a **merger** of the two argument structures. The two agent arguments are identified as the same, following a process of argument identification. The merger results in a composite argument structure which can be mapped further to grammatical relations such as subject and object at f-structure via Linking Theory (Butt, 2006).

DO <agent %Pred >

PINCH <agent, theme >

Figure 5.1: a-structures for 'do' and 'pinch' describe the two pieces of the NVC preceding the process of Argument Merger.

The notion of the light verb as co-predicator is also implemented in the corresponding computational grammar for Urdu (Butt and King, 2002). However, rather than posit a different level of representation (such as a-structure described above), the computational analysis stays at the level of f-structure. In order to compose the argument structure from the two co-predicators, a

Restriction Operator is used to manipulate the f-structure(Butt et al., 2003). This may be shown with an example from Butt et al. (2003) in Figure 5.2, which shows a simple f-structure for the lexical item 'Nadya'.

$$\begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \\ \text{CASE} & \text{erg} \end{bmatrix} \begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix}$$

Figure 5.2: The restriction operator has the ability to restrict out information e.g. the CASE feature from the attribute-value matrix (AVM) for the lexical item 'Nadya'. (↑/CASE) gives us the restricted second AVM in this figure.

The ability to restrict certain features allows the manipulation of the f-structure of predicate subcategorization frames. It also helps to capture certain other properties of complex predicates, such as stacking. Complex predicates can be stacked such that a noun verb complex predicate can occur inside a verb-verb complex predicate (Example 50 adds a verb verb complex predicate to an NVC).

(50) nadya=ne haathi=ko pinch kar li-yaa
nadya.F.Sg=erg elephant.M.Sg=acc pinch do take-perf.M.Sg
'Nadya pinched the elephant (completely)'

In such cases, f-structures for each predicate are composed, taking into consideration any restricted features. Therefore, the computational analysis is able to model the productive nature of the NVC in Urdu, while maintaining the core ideas from the theoretical account.

In the Urdu ParGram grammar (Butt and King, 2002), complex predicates have been implemented in the form of a 'dependency bank', which allows the representation of phenomena like NVCs in a theory-independent format. Therefore, while the NVC representation has a theoretical model of co-predication within a-structure in LFG, in the computational grammar this is implemented in a more theory-neutral format (i.e the *triples* format (Crouch et al., 2012b)).

With this general description of the LFG analysis, we now turn to the specific Hindi examples.

In section 5.2, we discussed a class of nouns (class 'B'), which alternate with *kar* 'do' and *hu-* 'happen'. Second, we also discussed the number and gender agreement of the light verb with the predicating nominal. In the LFG analysis, a sentence like (51) gets a final f-structure like 5.3 (Ahmed et al., 2012). Recall that the nominal *yaad* 'memory' is one of those nominals that can optionally take either accusative or nominative marked non-subject arguments. In example 51, the nominative argument is *sabaq* 'lesson' and the light verb shows default masculine, singular agreement (rather than showing agreement with the feminine *yaad*).

(51) naḍia=ne sabaq=ko yaad ki-ya
Nadia.F.Sg=Erg lesson.M.Sg=Acc memory.F do-perf.M.Sg
'Nadya remembered the lesson'

The f-structure in Figure 5.3 also shows the light verb *kar* and nominal *yaad* as co-predicators. The PRED feature shows the composed argument structure from the f-structures of nominal and light verb. Light verb *kar* subcategorizes for the agentive argument and another PRED viz. *yaad*. The nominal *yaad* itself takes a theme argument *sabaq*.[1] These have now been merged at the top level PRED.

Note that the SBCG or Sign-Based Construction Grammar approach is worth mentioning here as it is quite similar to the LFG analysis of the NVC (Hoffmann and Trousdale, 2013). For an example like (51), the light verb *kar* would have two arguments; agent and action (compare these to the LFG agent and PRED). The nominal *yaad* would also contain two arguments (agent and undergoer). The co-instantiation construction would equate the semantic indices of the two highest-ranking arguments from the two lists. Light verbs like *kar* would belong to the lexical class construction of co-instantiation (along with other transitive light verbs). The inchoative counterpart *ho* 'be' would not belong to this class, but take a single argument of its own.

[1]Note that while *kar* contains another predicate, the NVC and light verb are not examples of control structures such as *John tried to leave*. Butt (1995) shows that participial adverbials can only be controlled by the subject argument. This shows that there cannot be an embedded argument contributed by the preverbal element in the complex predicate.

In addition, the composed f-structure shows how features in the nominal's f-structure have been restricted. For example, we know that the light verb *kar* contributes an agentive argument. In order to code this lexical information, the feature `LEX-SEM` is used.The `LEX-SEM` feature is used to distinguish light verbs such as *kar* 'do' and *de* 'give' that require agentive arguments from those like *paR* 'fall', which do not. Further, the feature `VTYPE` on the nominal must be shown as complex rather than simple. Both features `LEX-SEM` and `VTYPE` are therefore restricted in the f-structure of the nominal so that they may be overwritten by the light verb.

Ahmed et al. (2012) also describe the case of *bahas kar* 'debate do; to argue' (52). In contrast to (51), the light verb agrees with the noun *bahas* and it appears in this f-structure as its object. At the same time, in Figure 5.4, the light verb *kar* is a co-predicator with *bahas*. The light verb's argument structure includes both the subject *meNdak* 'frog' and the nominal host *bahas* 'debate'. The only argument of *bahas* 'debate' is the oblique argument *bicchU* 'scorpion', post restriction.

(52) meNdak=ne bicchu=se bahas kii
frog.M.Sg=Erg scorpion.M.Sg=Inst debate.F.Sg do.Perf.F.S
'The frog argued with the scorpion.'

The individual feature structures for *bahas* 'debate' and *kar* 'do' are shown in Figure 5.5. The light verb's subcategorization frame includes the subject and an additional empty slot for the nominal predicate. In Figure 5.4, these are combined to form a single predicate. Note that the `CHECK` features are used to ensure the well-formedness of the grammar. Others such as `VTYPE` and `TNS-ASP` are self-explanatory. Ahmed et al. (2012)'s analysis does not describe the case of the same nominal alternating with the light verb *hu-* 'be' as in example 53. On the other hand, Ahmed and Butt (2011) mention that examples such as (53) are 'resultative state meanings', which probably implies that such cases are not considered NVCs in the LFG analysis and hence they are not described in the ParGram analysis.

(53) bicchu-se bahas hu-ii
scorpion.M-Inst debate.F.Sg be-Perf.F.S
'There was a debate with the scorpion; A debate with a scorpion happened'

$$\begin{bmatrix}
\text{PRED} & \text{`kar}\left\langle [\text{`nAdiyah'}] \text{ , `yaad}\left\langle [\text{`sabaq'}] \right\rangle \text{'} \right\rangle \text{'} \\
\text{SUBJ} & \begin{bmatrix}
\text{PRED} & \text{`nAdiyAh'} \\
\text{NTYPE} & \begin{bmatrix} \text{NSEM} & \left[\text{PROPER}\left[\text{PROPER-TYPE} \quad \text{name}\right]\right] \\ \text{NSYN} & \text{proper} \end{bmatrix} \\
\text{SEM-PROP} & \left[\text{SPECIFIC} +\right] \\
\text{CASE erg, GEND fem, NUM sg , PERS 3} &
\end{bmatrix} \\
\text{OBJ} & \begin{bmatrix}
\text{PRED} & \text{`sabaq'} \\
\text{CHECK} & \left[\text{NMORPH} \quad \text{obl}\right] \\
\text{NTYPE} & \begin{bmatrix} \text{NSEM} & \left[\text{COMMON} \quad \text{count}\right] \\ \text{NSYN} & \text{COMMON} \end{bmatrix} \\
\text{SEMPROP} & \left[\text{SPECIFIC+}\right] \\
\text{CASE acc, GEND masc, NUM sg, PERS 3} &
\end{bmatrix} \\
\text{LEX-SEM} & \left[\text{AGENTIVE} \quad +\right] \\
\text{TNS-ASP} & \begin{bmatrix} \text{ASPECT} & \text{perf} \\ \text{MOOD} & \text{indicative} \end{bmatrix} \\
\text{VTYPE} & \left[\text{COMPLEX-PRED} \quad \text{nv}\right] \\
\text{CLAUSE-TYPE} & \text{decl, PASSIVE -}
\end{bmatrix}$$

Figure 5.3: F-structure for the sentence *Nadya=ne sabaq=ko yaad kiyaa*

$$\begin{bmatrix}
\text{PRED} & \text{`kar'}\left\langle \left[\text{2:meNdak}\right], \text{`bahas}\left\langle \text{63:biccHU} \right\rangle \text{'} \right\rangle \text{'} \\
\text{SUBJ} & 2\left[\text{PRED} \quad \text{`meNdak'}\right] \\
\text{OBJ} & \left[\text{PRED} \quad \text{`bahas'}\right] \\
\text{OBL} & 63\left[\text{PRED} \quad \text{`biccHU'}\right]
\end{bmatrix}$$

Figure 5.4: Final (abbreviated) F-structure for the NVC *bahas kar* 'debate do'.Note that *bahas* acts simultaneously as a co-predicator and argument of the light verb

$$\begin{bmatrix} \text{PRED} & \text{`kar} \left\langle \text{[SUBJ], [-1]} \right\rangle\text{'} \\ \text{SUBJ} & \left[\text{CASE} \quad \text{erg}\right] \\ \text{OBJ} & \left[\text{GEND} \quad \text{fem, NUM sg}\right] \\ \text{CHECK} & \begin{bmatrix} \text{\_VMORPH} \left[\text{\_VTYPE} \quad \text{infl}\right] \\ \text{\_RESTRICTED - , \_VFORM perf} \end{bmatrix} \\ \text{LEX-SEM} & \left[\text{AGENTIVE} \quad +\right] \\ \text{TNS-ASP} & \left[\text{ASPECT} \quad \text{perf}\right] \\ \text{VTYPE} & \left[\text{COMPLEX-PRED-FORM} \quad \text{kar}\right] \\ \text{PASSIVE} & - \end{bmatrix} \begin{bmatrix} \text{PRED} & \text{`bahas} \left\langle \text{-OBL} \right\rangle\text{'} \\ \text{NTYPE} & \begin{bmatrix} \text{NSEM} \left[\text{COMMON} \quad \text{count}\right] \\ \text{NSYN common} \end{bmatrix} \\ \text{GEND} & \text{fem, NUM sg} \end{bmatrix}$$

Figure 5.5: F-structures for the light verb *kar* 'do' and the predicating nominal *bahas* 'debate' respectively

To summarize, the LFG analysis of NVCs is rooted in the f-structures of the lexical items. Both noun and light verb are presented as co-predicators, whose argument structures are merged in the final f-structure of the sentence. Moreover, the c-structure is factored apart from the f-structure, which gives the analysis some advantage with respect to syntactic flexibility. For example, the XLE implementation of the Urdu LFG grammar can parse *meNdak-ne bahas kii bicchu-se* 'frog-erg debate did scorpion-with; The frog debated with the scorpion '. [2]

The LFG analysis also models the difference between nouns like *yaad* 'memory' and *bahas* 'debate' via their lexical entries. Nominals like *yaad* 'memory' do not act as arguments as well as co-predicators like *bahas* 'debate'. Hence, their f-structures do not include a lexical entry for `OBJ` like *bahas*. At the same time, both noun and light verb are represented as predicates in the sentence.

5.4 Two approaches to NVC analyses

Two predicating elements that result in a single predication is a challenge for prevailing syntactic theories. When the verb alone is the main predicator in a sentence, all other elements are arguments or modifiers. If, however, there are two predicating elements, the argument structure is more complex. The linguistic literature on NVC representation shows two main divisions in the representation of NVCs.

The first approach described in the preceding section on LFG describes the noun and verb as co-predicators i.e. both noun and light verb contribute arguments. The second approach considers the noun in the NVC as the primary predicating element, where the light verb only assigns case to the verbal case-marked arguments in the sentence.A third approach also exists, where noun and light verb are concatenated together as a multi-word expression and together, they license the arguments in the sentence. While the first two options are worth exploring, we discard the third option for two reasons: first, the NVC is highly productive in Hindi, which would imply that

[2]However, it was unable to parse *meNdak-ne bahas bicchu-se kii* 'frog-erg debate scorpion-with did; 'The frog argued with the scorpion" i.e. when the nominal is separated from the light verb.

there will be individual representations for each noun and light verb combination in the grammar. Second, there is evidence that the NVC forms a phrasal category in the syntax (Mohanan, 1997; Davison, 2005). This means that individual components of the NVC may be moved away from each other, emphatic particles or negation may intervene and the noun component may be independently modified by an adjective. Therefore, the multi-word option would not be the best approach here.

Figure 5.6 shows the two options for composing the argument structure for an NVC like *ṯareef kar* 'praise do; praise'. The solid lines in the figures indicate the predicate-argument relation whereas the dotted lines show us how the separate argument structures were composed. In the noun-centric analysis, the light verb and nominal's argument structures are composed, but the verb contributes no arguments of its own. For the verb-centric analysis, the light verb contributes the argument *logon*, whereas the nominal contributes *pustak*.

The noun-centric analysis has been described in earlier work by Grimshaw and Mester (1988) and also more recently in the framework of Tree-Adjoining Grammar. Han and Rambow (2000) (henceforth HR) have analyzed the Sino-Korean complex predicate, which consists of a light verb *ha* or *hata* 'do' and a noun of Chinese origin. HR's analysis of the Korean NVC assumes that the nominal in the NVC is the true predicate and has the full array of syntactic arguments. The light verb on the other hand contributes no semantic information nor does it have any arguments of its own.

In the following section, we first extend a noun-centric analysis for Korean to Hindi NVCs (Vaidya et al., 2014). We then compare this analysis with the one already described in LFG to understand which approach might work better for Hindi NVC representation.

5.5 Lexicalized Tree-Adjoining Grammar

Tree-Adjoining Grammar (TAG) is a formal tree-rewriting system that is used to describe the syntax of natural languages (Joshi and Schabes, 1997). The basic structure of a TAG grammar is an elementary tree, which is a fragment of a phrase structure tree labelled with both terminal and non-terminal nodes. The elementary trees are combined by the operations of **substitution**

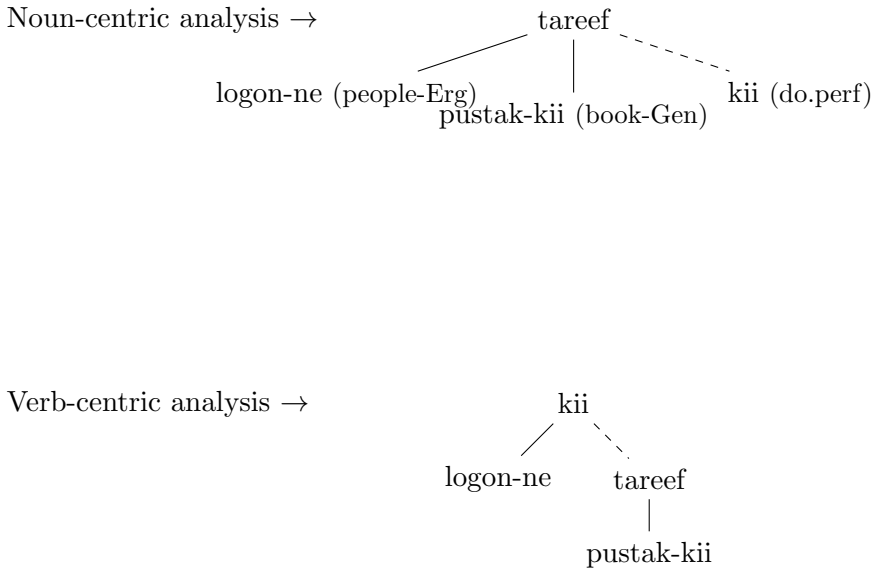

Figure 5.6: Derivation graphs showing two options for the analysis of *logon ne pus̱tak kii ṯareef kiii* 'People praised the book'. The LVC is *tareef kii*.

(where a terminal node is replaced with a new tree) or **adjunction** (where an internal node is split to add a new tree).

The elementary trees in TAG can be enriched with feature structures (Vijay-Shanker and Joshi, 1988). These can capture linguistic descriptions in a more precise manner and also capture adjunction constraints. TAG with feature structures is also known as FTAG (Feature-structure based TAG). A TAG can also be lexicalized i.e., an elementary tree has a lexical item as one of its terminal nodes. Lexicalized TAG enhanced with feature structures is known as Lexicalized Feature-based Tree-Adjoining Grammar (LF-TAG). This has been used for developing computational grammars for English (XTAG-Group, 2001), French (Abeillé and Candito, 2000) and Korean (Han et al., 2000). In our analysis, we will also use LF-TAG, but we will refer to it as LTAG for convenience.

Figure 5.7 shows the basic steps for composing elementary trees containing feature structures. Each node has a top and a bottom feature structure. Features can be shared among nodes in an elementary tree. In the tree for the verb *running*, the variable $\boxed{1}$ is used to show that the verb must share the same features as the subject NP.

The tree for *running* is an **initial tree** with a single terminal for its argument noun phrase (NP). The tree for *is*, on the other hand, is a special type of elementary tree called the **auxiliary** tree. It has a foot node (marked with an asterisk), which is identical to its root node. The auxiliary tree will adjoin into the tree for *running* at the VP node only. The top and bottom feature structures for MODE at the VP node for *running*, have different values (*ind*icative and *ger*undive), and they cannot unify. This captures an adjunction constraint for *obligatory adjunction* and requires adjunction to take place at this node only.

During adjunction, the top feature structure at VP_r in the auxiliary tree (for *is*) will unify with the top of the adjunction site (VP). The bottom feature structure at VP_r in the auxiliary tree will unify with the bottom of the adjunction site. During substitution, the top node in the tree for *Jill* unifies with the node at NP in the initial tree for *running*.

This results in the second tree in Figure 5.7, post the operations of substitution and adjunc-

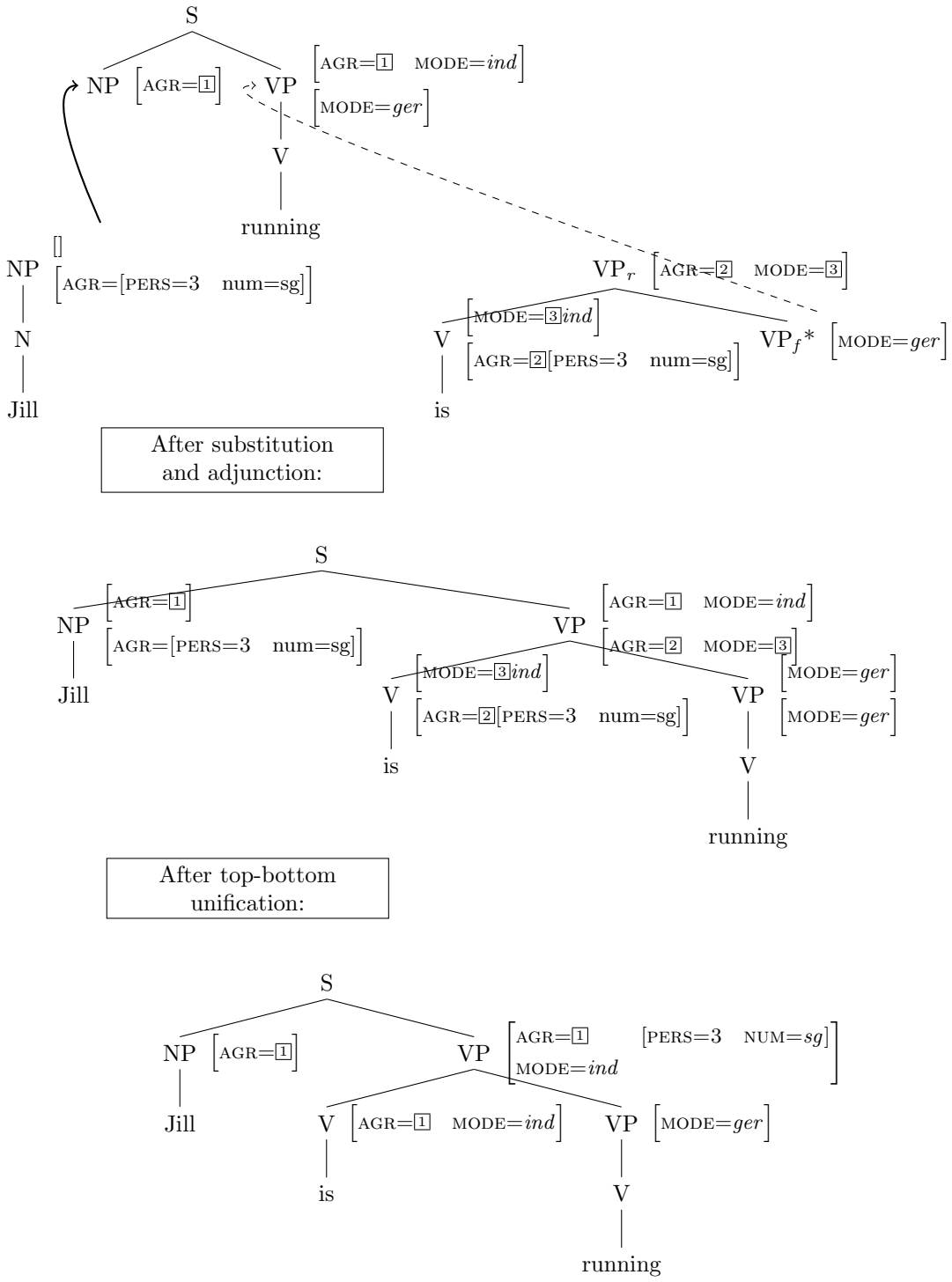

Figure 5.7: LTAG showing feature structures and constraints on adjunction (Example adapted from Kallmeyer and Osswald (2013)). The topmost trees show the operations of substitution (solid line) and adjunction (dashed line). Following these operations, we get a complete sentence 'Jill is running'. After both top and bottom nodes unify, the derivation is complete.

tion. In a final derivation step, top and bottom feature structures at each node will unify, to give the final derived tree with a single feature structure at each node. The resulting tree is called a *derived* tree.

An important characteristic of lexicalized elementary trees is their correspondence with the lexical item's predicate-argument structure. This has sometimes been formalized as the PACP (Predicate-Argument Co-occurrence Principle), which states that the elementary tree is the minimal syntactic structure that includes a leaf node for each of its realized arguments (Frank, 2002). The PACP restricts the structure of the elementary trees such that they may not be drawn arbitrarily. At the same time, lexicalized TAGs will often have the same lexical item realized as the anchor of varying syntactic realizations. For example a verb such as *run* will anchor a different elementary tree for its passive or interrogative variant.

5.5.1 Elementary trees for the Hindi NVC

In this section, we will adapt the Han and Rambow (2000) analysis of the Sino-Korean complex predicate for Hindi nominal predicates. In order to adapt the analysis to Hindi, we had to take into consideration the nominal's alternation with *kar* 'do' and *ho* 'be'. Additionally, we would also like to model the difference between nominals such as *tareef* 'praise' and *yaad* 'memory' with respect to the agreement facts (See 47).

Han and Rambow have proposed separate trees for the nominal and light verb. The elementary tree of the nominal is an an initial tree, and as it is considered the true predicate, it also chooses a syntactic structure that will realize all its arguments. The light verb on the other hand is represented as an auxiliary tree, therefore it is an adjunct to the nominals basic structure. However, as it is a predicate, it is also a special type of auxiliary tree viz., a predicative auxiliary tree (Abeillé and Rambow, 2000).

We also assume, following Han and Rambow that each node is specified with the feature `CAT` which has values like V or N, but the [`CAT=N`] feature on the noun is not realized unless the light verb composes with the elementary tree of the nominal. In addition, because the nominal is not a

verb, it has the feature `TENSE=-` i.e., it is not tensed.

The elementary tree for the nominal contains the full array of arguments for the NVC. In the LTAG framework, it is therefore a type of **initial** tree anchored by the nominal. The arguments of the NVC will be substituted into the tree in Figure (5.8).

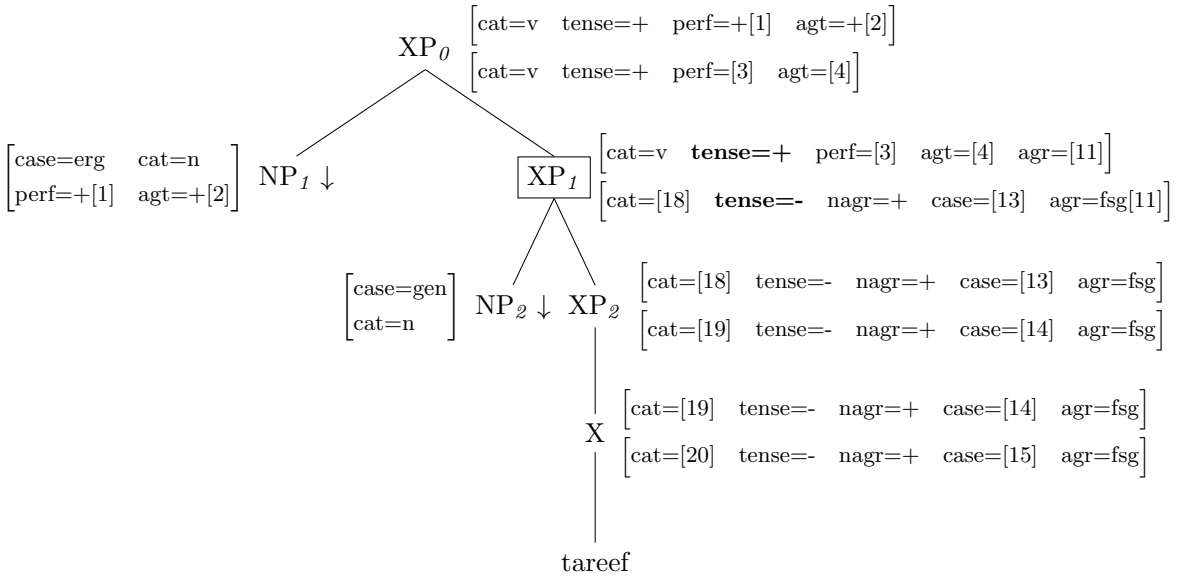

Figure 5.8: Tree for nominal *tareef* 'praise' (agentive), as seen in *logon ne pustak kii tareef kii* "People praised the book". The feature clash at XP_1 is marked with a box.

In order to model the light verb *kar* 'do' in Example 43, we will construct an **auxiliary** tree with feature structures, anchored at *kar* 'do'. Figure 5.9 shows such an elementary tree. The light verb *kar* is inflected for person, number, and gender as well as tense and aspect. In this particular example, it is tensed, feminine, singular and has perfective aspect; therefore it appears as *kii*. In Figure 5.9, the XP_r (root) node and its right-branching daughters are [`CAT=V`] with linguistic information about gender, number, tense and aspect. The feature `AGT=+` (agentive) at the top node implies that this auxiliary tree needs to unify with an initial tree that is also [`AGT=+`]. The XP_f (foot) node has [`TENSE=-`] and [`CAT=N`], which will enable it to adjoin into the elementary tree of a nominal. The `CASE` value is specified as `NOM` (nominative) as the light verb will assign nominative case to the noun.

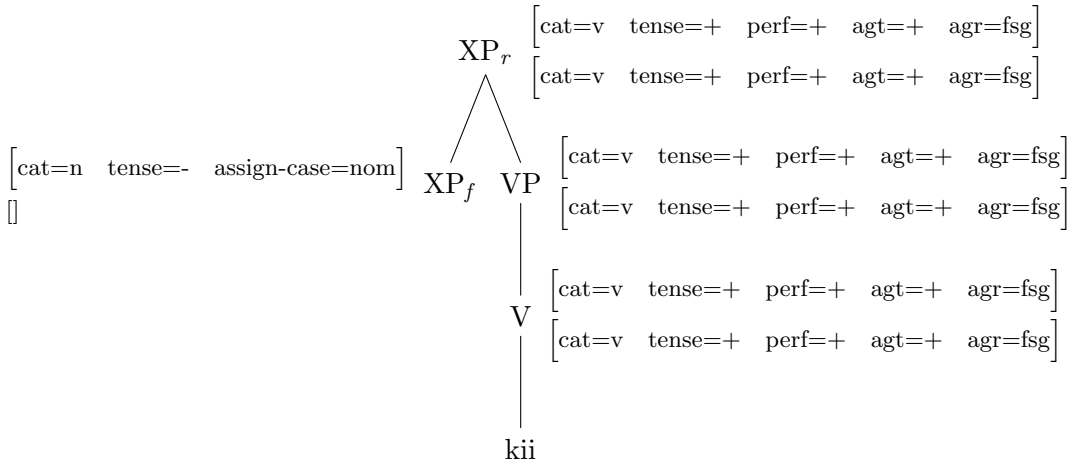

Figure 5.9: Elementary tree for light verb *kar* 'do' inflected as *kii*'do.fem.sing.perf'

The position of the arguments in the initial tree of the nominal roughly follows the configuration described in Bhatt et al. (2013)[p. 59], where the first position is the ergative-marked argument and is found in a transitive sentence [3] (but only if the property [PERF=+] is also present). The first position in Figure 5.8 has the features for [PERF=+] and [AGT=+] as a consequence of having [CASE=ERG]. The agentive argument shares the values for PERF and AGT with the S node. This ensures that the light verb that adjoins into this tree will match the PERF and AGT values in NP_1.

5.5.2 The nominal as an argument of the light verb

The light verb agrees in number and gender with the predicative nominal *tareef*. Therefore, it behaves as a predicating element and an argument simultaneously. In order to capture this syntactic property of the nominal predicate, our analysis takes advantage of the feature structures in LTAG. Additionally, we also manipulate the site of adjunction of the light verb in order to differentiate between nominals like *tareef* 'praise' and the non-agreeing nominals such as *yaad* 'memory'

We model this using the NAGR feature in Figure 5.8. The value for NAGR is positive as the light verb shows agreement with the nominal *tareef* for this example. Other arguments at NP_1 and NP_2 will not have an AGR feature unless they have nominative (null) case. When NAGR is positive, AGR gets its features from the nominal and these are passed up to the place of adjunction. When the light verb's agreement with the noun is not possible, e.g. if a higher nominative argument is available, the AGR feature will be populated by that argument.[4]

The feature clash at XP_1 between [TENSE=+] vs. [TENSE=-] shows us that the derivation of the tree is not yet complete. It forces adjunction to take place at this node because of the obligatory adjunction constraint in TAG. Post adjunction of the light verb, the XP_1 node is spliced into two subtrees, each dominated by CAT=N and CAT=V respectively (Figure 5.10). The NP_2 argument has

[3] Although ergative case can also be found in intransitive verbs that express volition e.g. *chiink* 'sneeze' or *cillaa* 'scream'. Intransitive NVCs with *kar* 'do' such as *snaan kar* will have elementary trees with the first position only, but this needs to be worked out further

[4] The agreement rule in Hindi also implies that more than one elementary tree will need to be proposed to account for the variation in the case realizations.

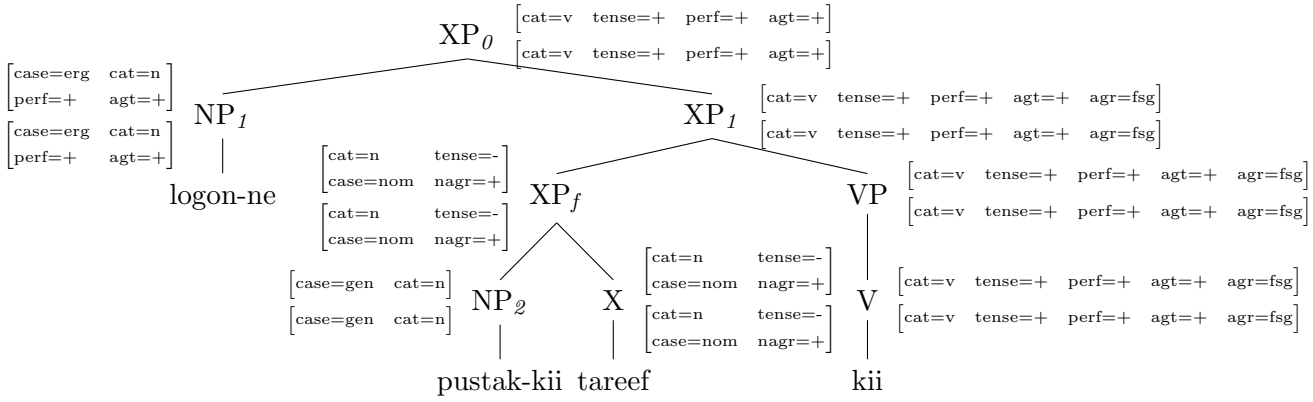

Figure 5.10: Post adjunction of the light verb's auxiliary tree into the initial tree *tareef* 'praise' at XP_2, we get the complete argument structure. Substitution at the nodes NP_1 and NP_2 gives us *logon-ne pustak-kii tareef kii* 'People praised the book'

genitive case, hence the node with `CAT=N` node dominates it. The light verb also assigns nominative case to the predicative nominal *tareef* 'praise'. If an NVC occurs with a non-subject argument that has non-nominal case (e.g. instrumental or locative, see example (52) for reference) then the elementary tree for those nouns will be slightly different. For such nouns, the site of adjunction will be below the non-subject argument, such that the `CAT=V` node dominates it (rather than `CAT=N`).

5.5.2.1 Exceptional nominals

Nominals such as *yaad* 'memory' form part of an exceptional class, where the light verb does not agree with the nominal (See 47). The light verb does not assign nominative case to the predicating noun *yaad* (Davison, 2005). In such cases, the value of `NAGR` is negative and adjunction into the tree of the nominal takes place at X (Figure 5.11). Recall that nouns such as *yaad* can optionally assign genitive case to the non-subject argument. However, when they do not (as shown in the tree in Figure 5.11), the nominal's sub-tree will not have arguments of its own. By manipulating the site of adjunction, we show that cases such as *yaad* can also be described by using the `NAGR` feature.

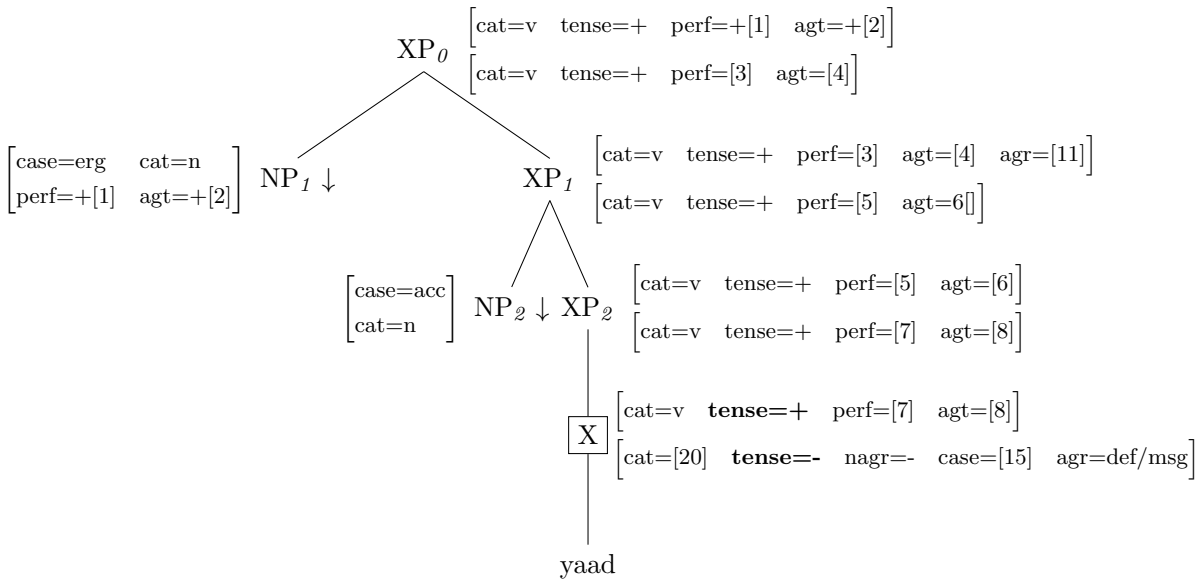

Figure 5.11: Tree for nominal *yaad* 'memory' (agentive), as seen in *Ram=ne Mohan=ko yaad kiyaa* "Ram remembered Mohan". The feature clash at X is marked with a box.

5.5.3 Alternation with *kar* and *ho*

The same noun *tareef* 'praise' may combine with the light verb *ho* (example 45). In that case, non-agentive *tareef* will choose an elementary tree such as Figure 5.12. This elementary tree appears without an agentive argument, hence NP_1 will have the feature `AGT=-`. Figure 5.12 shows that the site of adjunction into *tareef* 'praise' (non-agentive) is at XP_1. As the nominal is the only nominative argument , the light verb will agree with *tareef* (therefore `NAGR=+`).

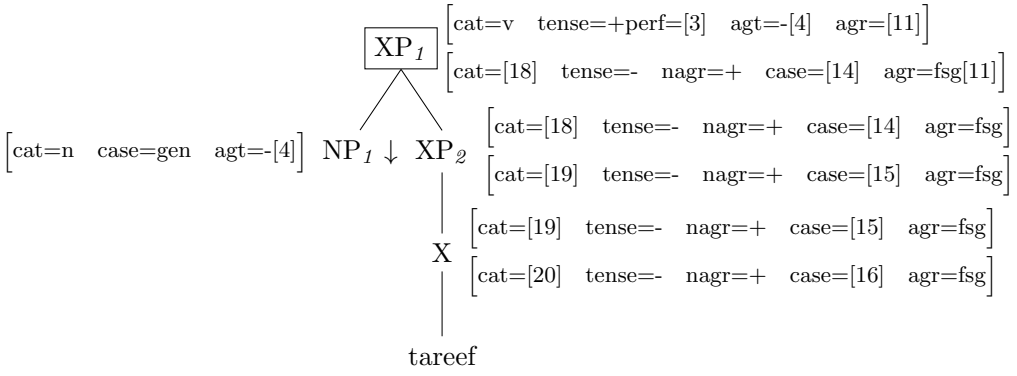

Figure 5.12: Tree for nominal *tareef* (non agentive) as seen in *pustak-kii tareef huii* '(the) book was praised'. The feature clash this time is at XP_1 and is marked with a box.

The tree for non-agentive *tareef* will always combine with a light verb that is `AGT=-`, in this case *ho* 'be'. In contrast with *kar* 'do', the auxiliary tree of the light verb *ho* 'be' will have [`AGT=-`]. The agent is implicit in the event described by this verb, and could have been shown in this tree as an empty category, but the presence of the feature `AGT=-` itself is illustrative of this fact.

5.6 Verb-centric vs. Noun-centric approaches

In section 5.3 and 5.5, we described two distinct analyses of the Hindi NVC using LFG and TAG. These analyses differ with respect to the representation of the light verb: in LFG, the light verb is a co-predicator (verb-centric analysis) whereas in TAG, it has no arguments of its own (noun-centric). Beginning from this broad distinction, we delve into a few other points of comparison in this section.

The first difference is based on the particular properties of the formalism used to represent NVCs. The LFG analysis requires a separate special operation to 'restrict out' i.e. prune and then merge the argument structures of the light verb. This is because any co-predication analysis will require argument merger i.e. linking or identification of corresponding arguments in two distinct argument structures to form a composite argument structure. In TAG, on the other hand, no special provision needs to be made in the analysis as the two argument structures are combined via the adjunction operation in the formalism. On the other hand, it is necessary to posit a large number of elementary trees for the nominal in TAG in order to capture all the syntactic alternations as well as differences in case realizations. As the elementary tree of the light verb does not contribute arguments of its own, separate elementary trees need to be posited for any alternations that can change the number and type of arguments in the NVC. TAGs in other languages, such as the English XTAG grammar are also known to have separate elementary trees for verbal alternations such as active and passive. Eventually, this increases the number of elementary trees in the grammar, hence the problem is not restricted to Hindi. Within the TAG framework itself, there have been attempts to reduce the number of elementary trees by using tree families, where all the syntactic alternations for a particular subcategorization frame are gathered together (Abeillé and Rambow, 2000).

The other point of comparison is with respect to the predicate-argument structures of noun and light verb. It is quite clear that the nominal contributes the non-subject argument in the sentence. For instance, in (55), the noun *zor* 'pressure' has a locative marked argument of its own (*us baat par* 'on that topic'), which is not available with the simple predicate *give* in (54).

(54) Ram=ne Mohan=ko kitaab d-ii
Ram.M.Sg=Erg Mohan.M.Sg=Dat book.F give.prf.F.sg
'Ram gave Mohan a book'

(55) Ram=ne us baat=par zor diyaa
Ram.M.Sg=Erg that topic.M.Sg=Loc pressure.M give.prf.M.sg
'Ram emphasized that topic'

It can be argued that the NVC's subject argument is also contributed by the nominal. If we look at the data from Korean NVCs, we might find additional insights about the behaviour of the NVC's subject argument. Korean also has a light verb like *kar*, e.g. 56, which can occur with two arguments *John* and *Tom* (Choi and Wechsler, 2001). These are understood as being selected by the Sino-Korean noun *tayhwa* 'talk' because they occur independently with the noun in example 57. In Hindi, the noun *baat* cannot take two genitive marked arguments, but can form a noun phrase with the two arguments of the NVC (see examples 59 and 58).

(56) john-i Tom-kwa tayhwa-lul ha-yess-ta
John-Nom Tom-with talk-Acc do-past-Dec
'John talked with Tom'

(57) John-uy Mary-wa-uy tayhwa
John-Gen Mary-with-Gen talk
'John's talk with Mary'

(58) john=ne mary=se baat kii
John.M-Erg Mary.F-Inst talk.F do.perf.F.sg
'John talked with Mary'

(59) john=se mary=kii baat
John.M=Inst Mary.F=Gen talk.F.Sg
'John's talk with Mary'

Therefore, one may argue that the nominal predicates that participate in the*kar* and *ho* alternation can subcategorize for two arguments. In that case, the representation of the elementary tree in Figure 5.8 is plausible. In the alternation with *ho*, the agentive argument is present (or assumed) but unexpressed. However, the key difference is the fact that in TAG, there is no argument *identification* i.e. even if the nominal subcategorizes for the agentive argument, it is not identified with the agentive argument of the light verb. The light verb is simply assumed to contribute nothing at all to the argument structure.

The other major point of difference is with respect to the alternation with the light verb *ho* for nominals like *tareef*. The LFG analysis maintains that these are resultative state readings,

and most likely are not NVCs. In the TAG analysis, it is maintained that the alternation with *ho* (or *hu-*) 'be' provides a useful lexical alternative to an alternative syntactic structure (such as a passive). The alternation of the light verb *ho* 'be' and *kar* 'do' is moreover a characteristic of a certain group of nominals only (not all can show this alternation e.g., *intizar* 'waiting' from the class 'C' type). Therefore, *tareef ho* 'praise happen' is analyzed as a light verb construction. The *kar* and *hu-* alternation is also a highly productive one as nearly 265 unique nouns in the Hindi Treebank show this alternation. These nouns constitute about 15% of all the annotated NVCs in the Treebank. Therefore, we do not consider this alternation as being a resultative state construction but an NVC.

A final point of comparison is with respect to the category of the nominal. In LFG, the predicating nominal is understood as a noun, whereas the TAG analysis assumes an under-specified category. The feature-values for `CAT` in the tree are used instead, and they are not fixed. This feature of the analysis is carried over from the Korean light verb analysis and helps in distinguishing the particular properties of the nominal as well as its syntactic status (as an agreeing or non agreeing type of nominal). The `CAT` feature, as well as the site of adjunction changes. The under-specification analysis was initially proposed for the nouns of Chinese origin that participate in forming Korean NVCs. Although the noun in the Hindi NVC is not necessarily a borrowed noun (although this is possible, see 49), one can argue that the nominal is not a regular NP i.e. it is a lexical category that forms a constituent with the light verb. Other analyses have also proposed that the predicating nominal is a type of *mixed category* analysis (Manning, 1993; Choi and Wechsler, 2001).

While both analyses can capture the NVC properties, the advantage of either might depend upon a given end use. While the TAG analysis relies on a large number of elementary trees and only one simple operation to combine them, it also suffers from a lack of generalization. In effect, for every alternation of the nominal with a light verb, a new elementary tree will have to be specified. If we were to choose a noun-centric design like TAG, however, it may be easier to link the nominal elementary trees with the nominal predicate frames described in Chapter 3. In TAG terms, this helps to determine how many substitution nodes must be created for a head in a TAG initial trees.

Elementary trees for adjuncts are created by converting subtrees of adjuncts and their heads into auxiliary trees with a foot node.

These elementary trees can play a useful role in extracting a TAG grammar from the Hindi Treebank Bhatt and Xia (2012). Elementary trees extracted from existing noun frames will eliminate an extra rule-writing step to extract predicate heads and their arguments from the Hindi Treebank. According to the mechanism described in Bhatt and Xia (2012), *e-tree based* rules are used to automatically elementary trees from the Hindi Treebank. Instead, it would be possible to utilize hand-corrected frame files for elementary tree generation. These, in conjunction with phrase-structure conversion rules could be used to extract TAG grammars from the Hindi Treebank.

The LFG analysis, on the other hand has a verb-centric approach that is more suited to the language facts for Hindi. In Butt (1995); Butt and Geuder (2001); Butt (1993), the status of the light verb has been described as a unique category that is distinct from that of an auxiliary or main verb. Specifically, light verbs have been shown to contribute a very specific linguistic meaning to a complex predicate, ranging from forcefulness to surprise or volitionality. Therefore, they cannot be represented as a meaning-less licenser of predication.

At the same time, the LFG analysis relies on argument identification (i.e. coindexing the subject argument of the light verb with that of the noun), followed by argument merger. The constraints of the LTAG formalism do not allow for an operation such as argument merger to take place. Such an operation necessitates the merging of two subcategorization lists, which is not possible within the elementary tree formalism in TAG. This implies that a modified TAG formalism may be required if such an operation were to take place. At the same time, an alternate 'verb-centric' analysis is possible, where elementary trees can be designed in a manner where the light verb contributes a single subject argument, and the nominal the non-subject argument. We have explored such an analysis in appendix B.

Finally, the issue of word order differences, where noun and light verb may be scrambled away from each other has not been addressed satisfactorily in either analyses. In TAG, this would

invariably result in an increase in elementary trees or require the use of a modified TAG formalism (Rambow and Lee, 2007). The XLE grammar in LFG could not parse a sentence when noun and light verb were scrambled away from each other. This problem remains to be explored for future work.

5.7 Discussion

In the beginning of the chapter, we outlined two challenges with respect to the syntactic representation of NVCs. The first of these, i.e. argument composition has been described in detail with respect to the noun-centric and verb-centric analyses in this chapter. The second challenge where the nominal simultaneously acts as a predicating element and as an argument of the light verb has been described using the f-structures for LFG and the feature strutures (i.e. the `NAGR` feature for TAG.

The design and creation of a lexicon is therefore crucial to develop a linguistically motivated analysis for NVCs. The representation of a productive phenomenon like complex predicates in lexicalized grammars is an important step in this direction. If we wish to capture the nuances of the NVC's argument structure, then the descriptive framework of lexicalized grammars can be used to provide a sound linguistic analysis as well as a computational implementation. Also relevant to the creation of the NVC lexicon is the decision to include or exclude certain phenomena (e.g the *ho* alternation in LFG) or whether to adopt a verb-centric or noun centric approach. The advantages and disadvantages of either approach have been discussed in detail in the previous section. A verb-centric approach has the advantage of capturing the linguistic facts for Hindi, but requires a formalism that can allow for an argument merger to take place. A noun-centric analysis for Hindi on the other hand can be of use in a grammar extraction experiment but lacks generalization. Either could be used or implemented, depending upon the end goal.

A secondary challenge with respect to lexicalized grammars, is to scale up the lexicon such that it can be used for computational applications. We described one such attempt for NVCs in Chapter 4. Such efforts are useful in order to utilize lexical grammars in computational applications.

Alternatively, it may be possible to use the informative lexicons of lexicalized grammars as features for a particular application. In a recent grammar extraction experiment for Hindi, a CCG lexicon was extracted from the Hindi Treebank. This was then used to extract informative lexical features to improve the performance of a Hindi dependency parser (Ambati et al., 2013).

There are several productive avenues that may be explored for future work. The first of these is with respect to implementing the NVC analysis for a large number of nouns in a lexicalized grammar framework (LFG or TAG). Second, is the potential to utilize these grammatical representations for improving applications such as parsing or semantic role labelling. Finally, phenomena like NVCs, with their two predicating elements challenge the architecture of the formalisms themselves and hence are a good test case for comparison across frameworks.

Chapter 6

Automatic Identification of NVCs

6.1 Introduction

The identification of NVCs is part of the larger problem of multi-word expression identification. The inability to accurately identify multi-word expressions is an impediment to effective Natural Language Processing (NLP). For instance, in the English Resource grammar, poor coverage of multi-word expressions was the cause of several parsing errors (Baldwin et al., 2004). Accurate detection of multi-words improves applications such as word sense disambiguation (Finlayson and Kulkarni, 2011) and machine translation (Pal et al., 2011). As Hindi NVCs are highly productive, identification of these expressions is necessary for more effective NLP applications.

In this chapter, we present our study on the automatic identification of NVCs in Hindi. We train a supervised classifier to identify NVCs based on the annotated Treebank data. This classifier is tested against a held-out set of NVCs and its performance is evaluated on these unseen test examples. We will first describe some relevant background literature on automatic identification of NVCs and following this, we describe our own experiments.

6.2 Previous Work

In the computational literature, common approaches towards the detection of multi-words such as NVCs include association measures, linguistic heuristics and other methods such as parallel corpora. Statistical association measures compute lexical associations between pairs of words. Knowledge intensive methods make use of linguistic knowledge in the form of resources (e.g. the-

sauri, dictionaries), heuristics or rules. A combination of one or more of these approaches is also possible.

6.2.1 Use of Association Measures

The use of a particular association measures depends upon the type of multi-word, the size of the corpora and its domain (Evert and Krenn, 2005). The pre-processing stage and the threshold used for excluding low-frequency multi-words also affect the type of association measure used. In the literature, log-likelihood (Dunning, 1993) has been used for detecting Hindi compound nouns (Kunchukuttan and Damani, 2008), mutual information (Church et al., 1991) has been used to detect English complex predicates (a.k.a. light verb constructions) (Fazly, 2007) and the t-score has been applied to German PP-verb pairs (Krenn, 2000).

In order to use lexical association measures, pairs of multi-word candidates are extracted from large corpora (This extraction step may require pre-processing of the corpus with syntactic or morphological information). These are then annotated with scores from a particular association measure (e.g. log-likelihood or $\chi 2$). The candidates with the highest scores are then inspected by a human expert or compared with existing gold standard lists for those multi-words. Before association measures are used, the candidate multi-words may also be ranked by frequency, as the most frequently occurring cases are usually good indicators of their multi-word status. Association measures are language and type-independent, i.e. they may apply to any language or any type of multi-word expression.

Ramisch et al. (2008)'s compared the language-independent association measures with language and type dependent approaches to multi-word detection. While they found association measures such as mutual information to be quite useful, they contrasted their performance with a linguistic measure termed "Entropy of Permutation and Insertion". This took into account the syntactic variants of an expression, and their results showed an improvement in the performance for the English phrasal verbs dataset. They concluded that while association measures are indeed useful, they need to be tuned towards the particular characteristics of the type of multi-word.

6.2.2 Use of Linguistic Knowledge

Linguistic knowledge in the form of dictionaries, thesauri and linguistic rules is used directly or formulated into a linguistic measure for the detection of multi-words. This knowledge is adapted towards the particular type of multi-word i.e. whether it is a phrasal verb, a complex predicate or a compound. These measures also take into account the peculiarities of the language that is being studied.

Vincze et al. (2011) compared the automatic detection of noun-noun compounds with NVCs. They found that the effect of syntactic information (such as dependency arc labels) is negligible for detecting compounds but drastically improves the performance in the case of NVCs. This shows that NVCs are a good test case for integrating statistical measures with linguistic information.

In English, complex predicates are more commonly termed as light verb constructions or support verb constructions (As we are referring to the same phenomenon, we will continue to use the term 'complex predicate'). Tan et al. (2006) have used a number of linguistic features to identify light verbs in English. They build on the work of Grefenstette and Teufel (1995) and Stevenson et al. (2004), who used features such as the morpho-syntactic similarity between nominal predicates and their verbal counterparts or the presence on indefinite determiners before the nominal predicate.

Tan et al. (2006) also make use of linguistic resources such as a list of light verbs that frequently occur in a complex predicate in English. Cross-linguistically, the same set of light verbs (such as *make*, *give*, *do/have*) tend to occur in a complex predicate. Additionally, in English, the noun in a complex predicate is very often the nominalized form of a verb. E.g. *take a walk* contains the nominalization *walk*, which also occurs as a full verb. This property is exploited in a feature that looks at the frequency of the nominalization and its full verb counterpart, i.e. *walk(v)* and *walk(n)*. If both tend to occur with the same frequency, then it is likely that the particular noun-verb candidate is a complex predicate. Using these features, Tan et al. use a random forest classifier to achieve an F1-score of about 0.68 on the Wall Street Journal corpus of English.

Tu and Roth (2011) look at both statistical and linguistic contexts to detect English complex

predicates. Among their local linguistic features, they utilize bigram information about the nominal head and light verb, the nouns themselves and the Levin verb class members of deverbal nouns. They achieve almost similar results using purely linguistic or purely statistical features - an accuracy of about 0.86. However, they also find that the linguistic features are more robust for minimal pairs where surface features are exactly the same, but only one of the pair is a true complex predicate. They also observe that mutual information and the noun-verb bigram feature give the system the greatest boost. In a more recent study, Chen et al. (2015) have described an improvement over Tu and Roth (2011)'s performance by using lexical features from WordNet, as well as word sense information. Using the Tu and Roth (2011) testset, they report a 0.89 F-score for English NVCs.

Author/Feature	**Tan et al'06** (Eng)	**Tu and Roth'11** (Eng)	**Begum et al'11** (Hin)
Deverbal noun	Y	Y	
Noun semantics	Y	Y	Y
Light verb list	Y	Y	Y
Presence post-posn			Y
Presence determiner	Y	Y	
Collocational measure		Y	Y

Table 6.1: Commonly used linguistic features for English and Hindi NVC detection.

Begum et al. (2011) have achieved an accuracy of around 0.85 for identification of Hindi complex predicates. In their approach,they use a Maximum Entropy classifier (Ratnaparkhi, 1998) with mostly linguistic features. Similar to English, they use the set of light verbs and the nouns themselves. Rather than check for the presence of the determiner (like English), they check for the presence of postpositions and demonstratives, which preferentially do not occur with a Hindi complex predicate. Like Tu and Roth, they use the verb-object bigram and the noun class information from Hindi WordNet (Narayan et al., 2002).

Table 6.1 summarizes some of the important linguistic features used by each of the supervised learning approaches described above. There are a few similarities in the linguistic features used, but each approach also tunes the features to the language in question, whether English or Hindi. Each approach also uses a parsed corpus and a lexical resource such as WordNet or VerbNet.

6.2.3 Other Methods

Apart from using linguistic or statistical measures with supervised learning, other methods such as parallel corpora have been used to extract multiwords. These methods are popular because a resource-rich language (like English) can be used to project linguistic information onto another language. Mukerjee et al. (2006) described the use of parallel corpora to detect Hindi complex predicates. English and Hindi data was automatically aligned and then POS (part of speech) tags were projected from English to Hindi. If the projected English tag was a verb and its Hindi correspondent was a noun or adjective, then they checked whether the word immediately following the noun was a light verb. If this was the case, the Hindi counterpart was tagged as a complex predicate. This idea was based on the intuition that a complex predicate in Hindi is likely to correspond to a single verb in English. A drawback of this method was the need for hand-correction of all the automatic alignments for better performance (this was mainly because of the quality of the parallel corpora). Sinha (2009) also used a similar method but with a lexicon of light verbs in Hindi and their translations in English.

6.3 NVC detection for Hindi

The previous work on NVCs demonstrates that the task of identification of NVCs in general has received a fair amount of attention. In this respect, our proposal to build an automatic classification system for NVCs describes a few more contributions. To begin with, we use our classifier in tandem with the insights gained from linguistic analysis of NVCs. We would like to explore whether the linguistic analysis lends itself to being encoded in the form of features. We also look more closely at some of the language-specific challenges for identifying NVCs. Additionally, we compare our model to Begum et al. (2011) with our features and demonstrate an improvement in performance.

In this section, we will first explore our data selection method, followed by a description of the candidate selection process. We use the manually annotated `POF` label as our source of knowledge

about the NVC. We describe the automatic classifier and our features. Finally we evaluate our performance using two test sets from different genres.

6.3.1 Data

The Hindi Treebank consists of news text, with a small sub-part consisting of conversational data, taken from fiction. The Treebank also provides Training and Testing splits for the news subsection. We decided to utilize the training/testing splits provided by the Treebank for our experiments.

The Treebank uses the `POF` label to annotate adjectives, adverbs as well as nouns. For our study, we chose to focus on nouns only, leaving out adjectives or other categories. We also focused on only the top 20 most frequently occurring light verbs in the corpus, which accounted for 90% of the cases annotated in the Treebank. These light verbs include the following: *kar* 'do', *ho* 'be/happen', *de* 'give', *hE* 'be', *raha* 'stay', *aa* 'come', *karaa/karvaa* 'cause to do', *lagaa* 'touch/feel', *jataa* 'convey', *le* 'take', *banaa* 'make', *rakh* 'keep', *chal* 'go', *uthaa* 'rise', *daala* 'put', *laDa* 'fight', *lag* 'seem', *ban* 'become', *maar* 'hit'. Each of these light verbs also appear as 'full' verbs i.e. they can also appear without a nominal predicate.

Predicate counts in the Hindi Treebank			
	All Predicates	**NVCs**	**NVC in this study**
Tokens	47163	16564	15057

Table 6.2: Nominal Predicates included in this study

Table 6.2 shows the distribution of the NVCs in the data and in this study. The reason for leaving out NVCs that did not occur with the top 20 light verbs was the possibility of increased annotation errors for these cases. Adjectives and adverbs were not included as we did not consider their properties as the primary focus of this work.

The distribution of the 20 light verbs in our training data is shown in Figure 6.1. It is apparent from this bar plot that the frequency of *kar* 'do' is the greatest, followed by *ho*, 'be' and *de* 'give'. The light verb *kar* 'do' has many more positive cases of NVCs as compared to negative

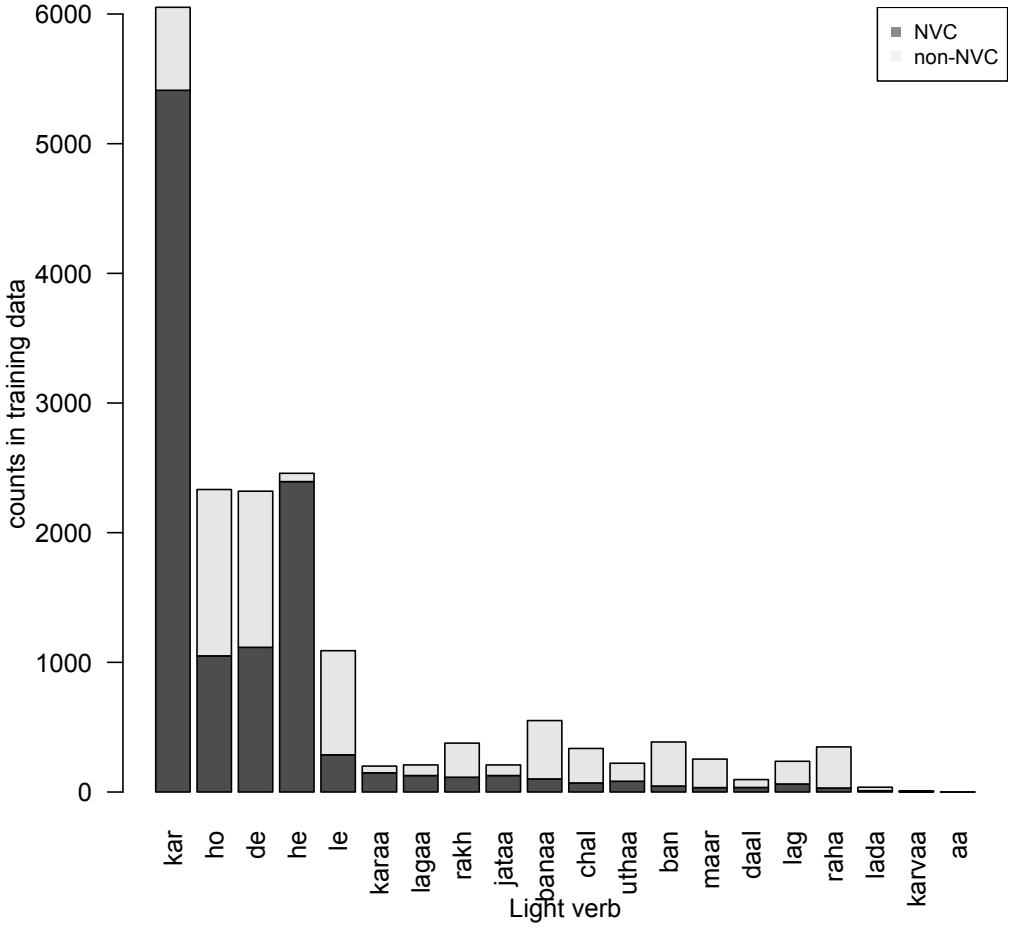

Figure 6.1: Light verb distribution in the training data.

or 'full' usages. For other light verbs such as *de*, the distribution is more even and with other light verbs, there are far more 'full' usages of these verbs as compared to light. This distribution of light verbs makes the overall detection of these verbs more challenging.

6.3.2 Candidate Selection

Our aim was to select positive and negative instances of NVCs in the Hindi parse trees. In the Treebank, the predicative noun is a dependent of the light verb and in the majority of the cases, both noun and light verb occur next to each other in the sentence. The noun and light verb can be scrambled away from each other, but we found this to be fairly rare in the Treebank NVC examples. Therefore, we chose candidates based on proximity- e.g. if a phrase annotated as an NP occurred next to a verb phrase containing a light verb, this was taken to be a candidate for NVC identification. If the predicative noun were to be scrambled away from the light verb, our system would not recognize it. The case of syntactically flexible NVCs needs to be taken up in future work.

Apart from NP phrases, we also accepted phrases annotated as 'BLK', which indicated that the noun was borrowed from English. Such nouns often occur as part of NVCs, as complex predication is used to introduce new words into the language.

6.3.3 Classifier

Our experiment is focused on the identification of NVCs in the Treebank. We use the LIBSVM implementation of a support vector machine classifier with a linear kernel for training as well as testing (Chang and Lin, 2011). We also utilized LIBSVM tools for scaling the data as well as for carrying out five fold cross validation during the development stage. The linear kernel requires the cost parameter c to be specified during the training stage. We utilized a c value 1 (default) for this data.

The training and testing instances consisted of positive and negative classes of NVCs. Table 6.3 shows the numbers of positive and negative labels in the training and testing data. Our training data was balanced in order to include a more or less similar number of NVCs and non-NVCs. We

did not change the test data.

Our two test sets were drawn from different genres. The news test set is from the testing portion of the Hindi Treebank. The second test set consists of sentences taken from literary criticism.This data is not from the Treebank, but taken from the ICON 2009 Shared task for Hindi parsing (Husain, 2009). We also included this dataset to compare the performance of our model with Begum et al. (2011),

It is noticeable that the news test set consists of a higher proportion of non-NVCs. This is in accordance with our earlier observations, where only 40% of the predicates are annotated as complex predicates. The literary criticism test data shows a similar trend, the non-NVCs occur with at a higher frequency as compared to NVCs.

We built two kinds of models using this classifier. The first of these was a combined model, where all twenty light verbs described in section 6.3.1 were tested against their non-light verb counterparts. The second type of model consisted of a separate classifier for the three most frequently occurring light verbs *kar* 'do', *ho* 'be' and *de* 'give'. We used the same training data for both types of models.

	Training (Hin Treebank)	**Test (News) Hin Treebank**	**Test (Literary) ICON Data**
NVC	9097	777	1008
Non-NVC	8993	857	1583

Table 6.3: Instances of NVCs and non-NVCs in the training and test datasets.

6.3.4 Features

NVC identification requires us to distinguish predicative noun usages from non-predicative ones. A non-predicative usage would be similar to an ordinary argument but a predicative usage would allow for combination with a particular light verb. There is no clear set of diagnostics that can be used to distinguish NVCs from non-NVCs. Bhattacharyya et al. (2007) discuss some diagnostic properties e.g the absence of postpositions after nouns that occur as part of NVCs. We include

some of those diagnostic criteria in this work and we also introduce some new features. While some of the morpho-syntactic evidence for NVC detection (e.g. postpostion information) can be very useful, it is not always available in a reliable manner. We need to include other information as well.

The usefulness of the lexical item feature has been mentioned earlier for NVC identification in English, therefore we include lexical information for both noun and light verb. Additionally, we also utilize collocational features using two association measures. Finally, we also include semantic features that describe the properties of predicative nominals.

Thus, the features used for identification of NVCs can be grouped into roughly four categories viz. lexical, morphosyntactic, collocational and semantic. Table 6.4 shows the set of features used for identifying NVC cases in the Treebank. We introduced feature number 4 and 8 based on our study of NVCs in Chapter 6 and Chapter 4. The remaining features have been used before to identify NVCs either in English or Hindi.

Type	**No**	**Feature**	**Description (Count)**
Lexical	1	Verb lemma (Baseline)	Feaure ID (20)
	2	Noun	Feature ID (2613)
Morpho-syntactic	3	Postposition after noun	Binary
	4	Arguments of eventive noun (eventive nouns have an 'extra' argument)	Binary
Collocational	5	Log-likelihood value	Numerical, converted to binary
	6	Pointwise Mutual Information value	Numerical,converted to binary
Semantic	7	Ontological category of noun	Feature ID (79)
	8	Acceptability of noun with a given light verb	Binary

Table 6.4: Features used for NVC detection

Lexical features

Lexical features are based on the intuition that predicative nouns and light verbs already found in training data are likely to be found in unseen data as well. We decided to use the verb lemma feature as our baseline. The verb and noun lexical features were represented as unique features i.e. each of the 20 light verbs had a feature id associated with it. Similarly, for nouns, each noun that occurred in the data was associated with a feature id. In Begum et al. (2011)'s experiments, lexical features alone resulted in nearly 81% accuracy. We expect to find a similar

effect for this feature in our data.

Morpho-syntactic features

We have two morpho-syntactic features viz. the postposition and extra argument feature. The first of these is based on the linguistic knowledge that a noun, which is part of an NVC cannot occur with a postposition. Example (60) shows that addition of a postposition after the noun in an NVC is ungrammatical. However, this postposition is correct for an ordinary argument NP in (61). A morpho-syntactic feature such as the presence of the postposition (we specifically look for genitive, locative or instrumental postpositions) is easy to extract from the Hindi Treebank using part-of-speech tags for postpositions and demonstrative pronouns.

(60) logõ=ne pustak=kii tareef=*ko ki-yaa
People.M.Pl=Erg book.F.Sg-gen praise.F-*acc do-Perf.M.Sg
*People praised the book

(61) samir=ne us seb=ko khaa-yaa
Samir.M.Sg=Erg that apple.M.Sg=Acc eat-Perf.M.sg
'Samir ate that apple'

The 'presence of extra arguments' feature looks for certain postpositions that indicate whether a nominal has introduced an argument of its own (apart from itself and the subject, which is introduced by the verb). Extraction of these features is facilitated by the dependency tree information.

(62) pulis=ne **logon=par** hamlaa ki-yaa
police=Erg people=loc attack.M.Sg do-Perf.M.Sg
'(The) Police attacked the people'

(63) samir=ne **mohan=se** nafrat k-ii
samir.M.Sg=Erg Mohan.M.Sg=instr hatred.F do-perf.F.Sg
'Samir hated Mohan'

(64) samir=ne **ghadii=kii** chorii k-ii
Samir.M.Sg-Erg watch.F.Sg-gen theft.F do-Perf.F
'Samir stole the watch'

Examples 62-64 illustrate the cases where an extra argument, with either genitive, accusative or locative case [1] is introduced by the nominal. We also experimented with using other part-of-speech features here- such as the presence of proper nouns (the NNP tag in the Treebank), but they were not useful, perhaps because they did not occur so often in the data. We also tried using the NN tag itself as a feature, but it was not very informative.

Collocational features

Collocational features are based on the idea that a particular noun and light verb combination is likelier to occur together rather than separately. In other words, we want to know whether a noun and verb are occurring together more often than chance. We decided to use the log-likelihood ratio that is known to be useful for sparse data (Manning and Schütze, 1999). In addition, we also chose to use pointwise mutual information (PMI), which has been used successfully for English NVC detection (Tu and Roth, 2011; Fazly, 2007).

In order to collect our counts for the collocational features, we had to make use of a larger corpus than the Treebank. We made use of Corpus Factory for Hindi consisting of 60 million words of web text, including Hindi Wikipedia (Kilgarriff et al., 2010). This corpus was tagged with part of speech information and proximate noun and verb combinations were extracted from it. We then made use of the N-gram statistical toolkit (Pedersen et al., 2011) in order to calculate the values for log-likelihood and pointwise mutual information.

These values were then converted to binary features. For log-likelihood, this was done using a table of critical values to decide whether the ratio was significant. Accordingly, it got the binary feature 0 or 1. In the case of PMI, we checked whether its value was greater than or less than 0 for a given noun and verb candidate. If it was greater, then the noun-verb pair was likely to be a better collocation.

Semantic features

The semantic features help us distinguish predicative nouns from their non-predicative coun-

[1] The locative and instrumental case might be ambiguous between an argument and adjunct

terparts. This was based on the idea that there are particular distinguishing semantic properties for predicative nouns e.g. many are eventive nouns rather than nouns describing artifacts or objects We used Hindi WordNet (Bhattacharyya, 2010) to look up feature 7 (the ontological feature). We first looked for the the synsets for a given noun candidate. If a synset for this noun existed, we extracted its ontological feature for the first sense of that noun. The ontological feature corresponded to properties such as *Abstract, Inanimate, Noun* or *Object, Inanimate, Noun* etc. Each of these were coded in the manner of lexical features i.e. a unique feature id for a particular ontological property.

The second semantic feature was the acceptability feature,which was designed based on the experiments in Chapter 4. We associated each of the 20 light verbs with the ontological property of the noun that was likely to occur with it. E.g. *kar* was associated with *Physical_Action_Abstract_Inanimate*, *de* 'give' with *Communication_Action_Abstract_Inanimate* etc. If a noun occurred with the ontological property that was associated with a particular light verb, it was marked positively for this feature. We had 20 unique feature ids for the acceptability property.

6.4 Evaluation

We trained our model on the eight features described above and evaluated them against the two test sets that we described earlier. We decided to make use of the verb lemma as our baseline feature. In addition to the evaluation on NVC and non-NVCs as a whole, we also evaluated our performance over individual light verbs in the news set. A detailed description of the results for each of these sets is described below.

6.4.1 Results: News testset

The set of eight features described earlier were used to create a model that was evaluated against the News test set. Table 6.5 shows the Precision, Recall and F1 score for our model. The verb lemma gave us a fairly high baseline, but addition of other features did improve the performance. When we break down the relative contribution of each feature, in Table 6.6, we find

that the noun's lexical feature makes the most contribution towards the results. Other linguistic features improve the performance slightly more. The usefulness of the lexical feature has also been described in Tu and Roth (2011). As both the training and test sets are drawn from the same distribution, similar nouns will tend to occur in both sets.

	Precision	**Recall**	**F1**
NVC	86.80	90.60	88.66
Non-NVC	91.13	87.51	89.28
Accuracy	88.98		
Verb lemma Baseline	80.59		

Table 6.5: Results for the news dataset from the Hindi Treebank

Feature combination	Accuracy	Diff
Baseline (verb lemma)	80.59	
Lexical	87.33	+.6.74
Lexical+Morphosyntactic	88.31	+ 1.02
Lexical+Morph+Collocational	88.98	+ 0.67
Lexical+Morph+Coll+Semantic	88.98	-

Table 6.6: Relative contribution of features for News dataset

We also decided to evaluate this model's performance on the individual light verbs in our data. We calculated the precision, recall and F1 score for each light verb that occurred in the test set. The results are described in Table 6.7. We find that the combined model (i.e. the model for all light verbs) is effective for certain light verbs, specifically, *kar* 'do' and *le* 'take'. However, the performance is not as effective for other light verbs, noticeably, *de*. The model is unable to detect NVCs when the light verbs are very infrequent e.g. verbs such as *raha* 'stay' or *banaa* 'make'.

These numbers also show that despite the very high baseline for a light verb like *kar* 'do', the linguistic features result in some improvement.

It is possible that light verbs other than *kar* and *ho* have properties that are quite different, therefore the same set of features are not useful for their identification. In order to check whether the performance for these individual light verbs might improve after addition of a different set of linguistic features, we decided to create individual models for the light verbs *de* 'give' and *ho* 'be'.

Individual LVs	Counts	Precision	Recall	F1	Baseline	Accuracy
kar 'do'	546	96.31	98.80	97.54	92.12	95.42
ho 'be'	191	75.25	72.27	73.73	47.12	72.77
de 'give'	205	59.37	69.51	64.04	60.00	68.78
le 'take'	88	53.84	53.84	53.84	85.2	86.36
lagaa 'touch'	48	66.66	84.21	74.41	60.41	77.08
jataa 'convey'	15	77.77	63.63	70.00	26.67	60.00

Table 6.7: Precision, Recall and F1 for individual light verbs in the news test set.

6.4.2 Individual models

The eight features described in Table 6.4 that were used in our combined model were not as effective for certain light verbs. We decided to experiment with creating individual models for two light verbs *de* 'give' and *ho* 'be'. We used the same training and test data, but a different set of features for these two light verbs.

In order to improve the performance, we introduced two new morpho-syntactic features and one new collocational value. The first feature was the presence of a demonstrative pronoun before the noun. This indicates that it is not an NVC, but an ordinary argument. We detect this using the part of speech tag for such pronouns. The second morphosyntactic feature was based on the dependency tree structure. If a noun is represented with a dependent, it is likely to be a predicative noun. This feature overlaps with the 'extra argument' feature, but takes into account additional parse tree information about the predicative nature of the noun. (Figure 6.2).

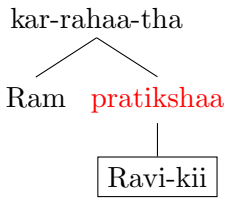

Figure 6.2: Tree for *Ram Ravi-kii pratikshaa kar rahaa thaa* 'Ram was waiting for Ravi'. The NVC is *pratikshaa kar* and the noun *pratikshaa* has a dependent

In addition to the morpho-syntactic features, we also utilized the LVC collocational value introduced in (Fazly and Stevenson, 2007). This has three factors- the relative frequency of a

noun N in the corpus, the probability with which a noun forms an LVC with any verb and the probability of forming an LVC with the given verb only. We calculated this using the Treebank knowledge about NVCs.

$$LVC(N,V) = Pr(N, LV, LVC) = Pr(N)Pr(LVC|N)Pr(LV|N, LVC)$$

Using these additional features, we created an individual model for the light verbs *ho* 'be' and *de* 'give'. Table 6.8 shows the performance of these individual models with more customized features. The individual models for *de* 'give' and *ho* 'be' show an improvement over the combined model with a different feature set. For *ho*, we enhanced the original eight features with the two morphosyntactic features.

We found that including semantic features for the light verb *de* 'give' actually resulted in a *decrease* in performance. The individual model for *de* 'give' consisted of morphosyntactic, collocational and lexical features only. We think this could be because the light verb *de* 'give' has specific semantic properties that are not captured by the semantic features we used for the combined model.

LV	**Model**	**Precision**	**Recall**	**F1**	**Baseline**	**Accuracy**
ho	Combined	75.25	72.27	73.73	47.12	72.77
	Individual	85.22	72.25	79.36	47.12	79.58
de	Combined	59.37	69.51	64.04	60.00	68.78
	Individual	69.41	71.95	70.65	60.00	76.10

Table 6.8: Comparing the performance of individual models with the combined model for *de* 'give' and *ho* 'be'

6.4.3 Results: Literary criticism

We also tested our combined model on a test set drawn from a different genre as compared to the training set. On the literary criticism test set, we obtained an accuracy of 86.41% (Table 6.9). Here, the lexical feature contributed to a much lesser extent as compared to the previous dataset as the nouns in the test set are from a different distribution. Therefore, the baseline accuracy based on the verb lemma feature is also lower. We also compared the performance of the classifier in this section with the study in Begum et al. (2011). In their study, a smaller training set from

the Hindi Treebank was used as a training set. Similar to our study, the twenty most frequently occurring light verbs made up the candidate NVCs. The test data was taken from a different genre, i.e literary non-fiction. We report some improvement over the accuracy in Begum et al. (2011), who used the same test set.[2]

The recall for the NVC cases is not as high as the News dataset, hence we notice a drop in the F1 score for the NVC cases. On the other hand, the precision and recall for the non-NVC cases is as good as the results from the news dataset.

	Precision	**Recall**	**F1**
NVC	86.66	76.98	81.5
Non-NVC	86.31	92.41	89.2
Accuracy	86.41		
Begum et. al. (2011)	85.28		
Verb lemma Baseline	76.76		

Table 6.9: Results for the literary criticism dataset from the shared task. We compare our result to the accuracy reported in (Begum et al., 2011)

If we look at the relative contribution of each feature in table 6.10, we can see that linguistic features can help in improving the performance when the test set is from a different genre. The relative contribution from the lexical feature is also lower as compared to the news test set.

Feature combination	**Accuracy**	**Diff**
Baseline (verb lemma)	76.76	
Lexical (noun lemma)	82.09	+5.33
Lexical+Morphosyntactic	83.94	+1.85
Lexical+Morph+Collocational	85.48	+1.54
Lexical+Morph+Collocational+Semantic	86.41	+0.93

Table 6.10: Relative contribution of features for literary criticism dataset

6.5 Error Analysis

In order to carry out a detailed error analysis, we focused on the number of unseen nouns that the classifier was able to predict accurately. As expected, the news testset had a fewer number

[2]Begum et. al. do not report precision, recall and F1 scores for their experiment.

of unseen nouns as compared to the literary testset. The classifier was able to predict about 86% of the unseen nouns correctly in both the literary and news test sets. In the literary testset, the majority of the errors were related to the prediction of the correct NVC cases—the classifier was better at correctly predicting those unseen nouns that were not NVCs. Some unseen NVCs in the literary testset were rare and the statistical or semantic features were not able to capture them. In such cases, the morpho syntactic features were able to predict the non-NVC cases with better success.

In the news test set, the classifier was about as good at predicting NVCs as well as non-NVCs among the unseen nouns. At the same time, a number of already seen nouns in the news test set were also not classified correctly. Some of these cases showed a more exceptional syntactic pattern e.g. some of these were non-finite light verbs such as gerunds, or were cases where the light verb had no dependent other than the predicating nominal. But apart from these cases, there was no over-arching pattern that could describe why nouns that were already seen in the training data could not be correctly identified in the test data.

	News Testset		**Literary Testset**	
	Unseen Nouns	Seen Nouns	Unseen Nouns	Seen Nouns
Correctly Predicted	156	1298	528	1711
Incorrectly Predicted	24	156	84	268
Total	180	1454	612	1979

Table 6.11: Unseen nouns in News and Literary criticism testsets

6.6 Discussion

The identification of Hindi NVCs appears easy enough at first because of the high performance of the lexical features. Almost all previous work describes the usefulness of this feature. However, if we evaluate the data by light verb, we find that there is variation in the performance across individual light verbs. This points towards the fact that the combined model for NVCs is not necessarily robust across the board. For light verbs that have markedly different linguistic

properties, or those light verbs that occur with low frequency, the combined model is not useful. For such cases, individual models are more useful, provided there is enough training data.

To the best of our knowledge, previous evaluation criteria do not examine the performance of a combined model for NVCs against individual light verbs. In Hindi, such an evaluation is necessary because of the peculiarities of the light verb distributions and the difference in their properties. For example, we find that semantic features in the combined model are not sufficient to capture the peculiarities of the light verb *de* 'give'.

This work also gives us an insight into the way a better NVC identification system might be designed. Such a system might use varying strategies for identification depending upon the light verb that occurs in an NVC. We could make use of individual models to identify high frequency NVCs with better accuracy. For low frequency cases, collocational features can be applied and finally, lexical features alone can be used to capture any NVC cases that were not identified by any other strategy. Such a system should result in an improved performance in a linguistically motivated manner.

Chapter 7

Conclusion and Future Work

7.1 Summary and Contributions

The work presented for Hindi NVCs is a step forward towards understanding this phenomena in an empirical setting. We began our research into NVCs with the Hindi Treebank, where nearly 37% of the predicates are annotated as NVCs. The productivity of NVCs and the challenges with respect to their linguistic representation and accurate identification are addressed in this thesis.

In the first part of this work, we described the semi-automatic creation of semantic frames for NVCs (Vaidya et al., 2013). These specify the subcategorization information for NVCs in terms of PropBank style semantic roles. This work is based on a detailed study of the mapping between the dependency labels in the Hindi Treebank and the PropBank semantic roles (Vaidya et al., 2011; Vaidya and Husain, 2011). Our mapping rules can identify almost two thirds of the semantic roles for NVCs correctly, which reduces the manual effort required to create these frames.

The large number of NVC combinations also raise problems with respect to consistency in annotation and representation in a lexical resource. The discovery of generalizations among nouns is useful in predicting similar semantic frames for a given group of nouns. In chapter 4, we described the use of a clustering algorithm to find similarity among nouns using two features. These results, though preliminary are useful directions for future work. We also showed how the results of such noun class groupings can be used to extend the lexicon for computational grammars (Sulger and Vaidya, 2014).

We continued our exploration of lexicalized grammars by comparing the representation of

NVCs in two frameworks viz., Lexical Functional Grammar (LFG) and Lexicalized Tree-Adjoining Grammar (TAG). This comparison highlights the differences between the two formalisms and the mechanism required to capture the syntactic properties of two light verbs *kar* 'do' and *ho* 'be'. While the LFG analysis captures the co-predicating properties of noun and light verb, the TAG analysis is can be used to extract elementary trees from the Treebank.

The final part of the thesis uses the insights gained about NVC properties in order to automatically identify these cases in the Treebank. We use a suite of eight features which identify NVCs across two genres. In addition, we also demonstrate that the use of individual models for a particular light verb can be used to identify NVCs with greater accuracy. This procedure also takes into account the peculiarities in the distribution of these light verbs in corpora.

We summarize our contributions to the study of NVCs below:

Development of semantic frames for Hindi NVCs Our method resulted in the creation of 1884 frame files for nearly 3000 unique combinations of noun and light verb. These frames are already in use for semantic role annotation of NVCs.

A Tree-adjoining grammar representation for a sub-class of Hindi NVCs

We modelled the linguistic properties of the Hindi NVC using Lexicalized Tree-adjoining grammar (LTAG). We use LTAG to describe the particular properties of NVCs using feature structures in LTAG. Additonally, we compare our analysis to an existing Lexical Functional Grammar analysis in (Ahmed et al., 2012). Our work contributes towards a theoretical understanding of NVCs in computational formalisms and can be used in future experiments on grammar extraction out of Treebanks.

Automatic identification of NVCs in corpora

We have built and evaluated a system for the automatic identification of NVCs. We have demonstrated the usefulness of our linguistic features and additionally, we have shown that features tuned to individual light verbs can improve the performance. Our system performs comparably with respect to previous work on NVC identification.

Mapping between Hindi Treebank and PropBank

We have developed a rule based system for mapping between the Hindi Treebank and PropBank labels, which can be used to improve tasks such as semantic role labelling and semi-automatic annotation of roles for Hindi.

Exploration of noun class detection for predicative nominals

We examined the contribution of two linguistic properties for discovering noun classes using a clustering algorithm. We also proposed a new evaluation measure based on PropBank semantic labels. Additionally, we described a procedure for integrating noun classes into Lexical Functional Grammar, using noun templates.

7.2 Future Work

In terms of future work, there are several promising areas that can be explored in more detail. The first of these involves more primary research on the properties of predicating nominals. As noted earlier, there are several exceptional predicating nominals that can optionally take accusative case marked arguments. A more detailed study of such nominals is required in order to uncover possible reasons for this exceptional behaviour. A psycholinguistic approach may be adopted to compare these nominals with other predicating nominals along semantic or discourse dimensions.

We also looked at the representation of NVCs in lexicalized grammars, and in particular we described the representation of NVCs using specialized templates. Another productive area of work might be to look at the design of such templates in other unification-based grammars such as HPSG or Sign-based Construction grammar.

For the automatic identification of NVCs, we described the performance of the model on individual light verbs. We also showed that the application of light verb specific features improved classifier performance. Therefore, it would be useful to understand whether models trained on each of the individual twenty light verbs might be used, as opposed to a single combined model. It may also be useful to use test sets from a number of different genres to check the robustness of the identification system.

The identification of true NVC cases themselves is still a challenging task. In chapter 2, we

described a gradient among NVCs. NVCs range from the most prototypical cases to the least, based on their performance on various diagnostic tests. It is worth exploring whether NVC annotation could be made similarly fine-grained for lexical resources. NVCs themselves could be annotated with more graded labels, reflecting the degree of compositionality between noun and light verb.

Therefore, there is potential for further research on NVCs at a primary level i.e. linguistic description, at the level of grammar engineering as well as in linguistically motivated NLP applications. For languages like Hindi, as well as other languages in South Asia, the NVC phenomenon cannot be ignored whilst working in any of the three areas described above. Given the productivity of NVCs, using a combination of linguistic and computational approaches is likely to yield the best results.

Bibliography

Abeillé, Anne. 1988. Light verb constructions and Extraction out of NP in TAG. In L. MacLeod, G. Larson, and D. Brentari, eds., Proceedings of the 24th Annual Meeting of the Chicago Linguistics Society.

Abeillé, Anne and Marie-Hélène Candito. 2000. FTAG: A Lexicalized Tree-Adjoining Grammar for French. In Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing. CSLI Publications.

Abeillé, Anne and Owen Rambow. 2000. Tree Adjoining Grammar: An Overview. In A. Abeillé and O. Rambow, eds., Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing. CSLI Publications.

Ahmed, Tafseer. 2010. The interaction of light verbs and verb classes of Urdu. In Interdisciplinary Workshop on Verbs - The Identification and Representation of Verb Features, Pisa.

Ahmed, Tafseer and Miriam Butt. 2011. Discovering Semantic Classes for Urdu N-V Complex Predicates. In Proceedings of the International Conference on Computational Semantics (IWCS 2011), Oxford.

Ahmed, Tafseer, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2012. A reference dependency bank for analyzing complex predicates. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12).

Alsina, Alex, Joan Bresnan, and Peter Sells. 1997. Complex Predicates: Structure and Theory. In Complex Predicates. CSLI Publications, Stanford.

Ambati, Bharat Ram, Tejaswini Deoskar, and Mark Steedman. 2013. Using CCG categories to improve Hindi dependency parsing. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 604–609.

Bahl, K.C. 1974. Studies in the Semantic Structure of Hindi, vol. 1. Motilal Banarasidass, Delhi.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics.

Baldwin, Timothy, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2004. Road-testing the English Resource Grammar over the British National Corpus. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, Portugal.

Barrett, Leslie and Anthony R Davis. 2003. Diagnostics for determining compatibility in English support-verb-nominalization pairs. In Proceedings of the 4th international conference on Computational Linguistics and Intelligent text processing (CICLing '03).

Begum, Rafiya, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency Annotation Scheme for Indian Languages. In Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP). Hyderabad, India.

Begum, Rafiya, Karan Jindal, Ashish Jain, Samar Husain, and Dipti Misra Sharma. 2011. Identification of Conjunct Verbs in Hindi and their effect on Parsing Accuracy. In In Proceedings of the 12th CICLing, Tokyo, Japan.

Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. 1995. Natural Language Processing: A Paninian Perspective. Prentice-Hall, India.

Bharati, Akshar, Rajeev Sangal, and Dipti Misra Sharma. 2007. SSF: Shakti Standard Format Guide. Tech. rep., IIIT Hyderabad.

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

Bhatia, Archna, 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.

Bhatia, Archna, Ashwini Vaidya, Bhuvana Narasimhan, and Martha Palmer. 2013. Hindi Propbank Annotation Guidelines. Manuscript.

Bhatt, Rajesh. 2008. Complex Predicates and Agreement. Presentation at EFLU, Hyderabad.

Bhatt, Rajesh and Elena Anagnostopoulou. 1996. Object Shift and Specificity: Evidence from ko-phrases in Hindi. In L. Dobrin, K. Singer, and L. McNair, eds., Papers from the Main Session of Chicago Linguistic Society 32, vol. 32.1, pages 11–22.

Bhatt, Rajesh, Annahita Farudi, and Owen Rambow. 2013. Hindi-Urdu Phrase Structure Annotation Guidelines. http://verbs.colorado.edu/hindiurdu/guidelines_docs/PhraseStructureguidelines.pdf.

Bhatt, Rajesh, 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.

Bhatt, Rajesh, Owen Rambow, and Fei Xia. 2011. Linguistic Phenomena, Analyses, and Representations: Understanding Conversion between Treebanks. In Proceedings of the 5th International Joint Conference on Natural Language Processing,Chiang Mai, Thailand, pages 1234–1242.

Bhatt, Rajesh, Owen Rambow, and Fei Xia. 2012. Creating a Tree-Adjoining Grammar from a Multilayer Treebank. In Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11), pages 162–170.

Bhatt, Rajesh and Fei Xia. 2012. Challenges in Converting between Treebanks: a Case Study from the HUTB. In Proceedings of META-RESEARCH Workshop on Advanced Treebanking held in conjunction with LREC-12.

Bhattacharyya, Pushpak. 2010. IndoWordNet. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), pages 3785–3792.

Bhattacharyya, Pushpak, Debasri Chakrabarti, and Vaijayanthi Sarma. 2007. Complex Predicates in Indian languages and Wordnets. Language Resources and Evaluation 40(3-4):331–355.

Bonial, Claire, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: Semantics of New Predicate Types. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).

Bresnan, Joan and Annie Zaenen. 1990. Deep Unaccusativity in LFG. In K. Dziwirek, P. Farrell, and E. Mejías-Bikandi, eds., Grammatical relations: A cross-theoretical perspective. CSLI Publications, Stanford.

Butt, Miriam. 1993. The Light Verb Jungle. In G. Aygen, C. Bowers, and C. Quinn, eds., Harvard Working Papers in Linguistics: Papers from the GSAS/Dudley House workshop on light verbs, vol. 9.

Butt, Miriam. 1995. The Structure of Complex Predicates in Urdu. CSLI Publications, Stanford.

Butt, Miriam. 2006. Theories of Case. Cambridge University Press.

Butt, Miriam. 2010. The Light Verb Jungle: Still Hacking Away. In M. Amberber, M. Harvey, and B. Baker, eds., Complex Predicates in Cross-Linguistic Perspective, pages 48–78. Cambridge University Press.

Butt, Miriam, Tina Bögel, Annette Hautli, Sebastian Sulger, and Tafseer Ahmed. 2012. Identifying Urdu Complex Predication via Bigram Extraction. In Proceedings of COLING 2012: Technical papers, pages 409–424.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshu Mashuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In Proceedings of the Workshop On Grammar Engineering And Evaluation.

Butt, Miriam and Wilhelm Geuder. 2001. On the (semi)lexical status of light verbs. In N. Corver and H. van Riemsdijk, eds., Semi-Lexical Categories. Mouton de Gruyter.

Butt, Miriam and Tracy Holloway King. 2002. Urdu and the Parallel Grammar project. In Proceedings of the 3rd workshop on Asian Language resources and International Standardization, vol. 12.

Butt, Miriam and Tracy Holloway King. 2007. Urdu in a Parallel Grammar Development Environment. Language Resources and Evaluation: Special Issue on Asian Language Processing: State of the Art Resources and Processing 41.

Butt, Miriam, Tracy Holloway King, and John T. Maxwell III. 2003. Complex Predication via Restriction. In Proceedings of the LFG03 Conference.

Butt, Miriam, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. A Grammar Writer's Cookbook. CSLI Publications.

Butt, Miriam, Tracy Holloway King, and Gillian Ramchand. 2008. Complex Predication:How Did the Child Pinch the Elephant? In L. Uyechi and L. Wee, eds., Reality Exploration and Discovery: Pattern Interaction in Language and Life. CSLI Publications, Stanford.

Butt, Miriam and Aditi Lahiri. 2013. Diachronic pertinacity of light verbs. Lingua 135:7–29. http://www.sciencedirect.com/science/article/pii/S0024384112002549.

Chang, Chih-Chung and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3):1–27.

Chen, Wei-Te, Claire Bonial, and Martha Palmer. 2015. English Light Verb Construction Identification Using Lexical Knowledge. In Proceedings of the AAAI-15, Austin, TX, USA.

Choi, Incheon and Stephen Wechsler. 2001. Mixed Categories and Argument Transfer in the Korean Light Verb Construction. In F. van Eynde, L. Hellan, and D. Beermann, eds., Proceedings of the 8th International HPSG Conference.

Church, Kenneth, William Gale, Patrick Hanks, and Donald Hindle. 1991. Using statistics in lexical analysis. In U. Zernik, ed., Lexical Acquistion: Exploiting On-line resources to Build a Lexicon, pages 115–164. Lawrence Erlbaum.

Claridge, Claudia. 2000. Multi-word Verbs in Early Modern English: A Corpus-based Study. Editions Rodopi B. V., Amsterdam-Atlanta edn.

Copestake, Ann, Fabre Lambeau, Aline Villavicencio, Francis Bond, Timothy Baldwin, Ivan A. Sag, and Dan Flickinger. 2002. Multiword Expressions: linguistic precision and reusability. In Proceedings of LREC 2002.

Crouch, Dick, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2012a. XLE Documentation. Palo Alto Research Center.

Crouch, Dick, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2012b. XLE Documentation. Tech. rep., Xerox Palo Alto Research Center.

Dalrymple, Mary, Ronald M. Kaplan, and Tracy Holloway King. 2004. Linguistic Generalizations over Descriptions. In M. Butt and T. H. King, eds., Proceedings of the LFG04 Conference. CSLI Publications.

Davison, Alice. 2005. Phrasal predicates: How N combines with V in Hindi/Urdu. In T. Bhattacharya, ed., Yearbook of South Asian Languages and Linguistics, pages 83–116. Mouton de Gruyter.

Dunning, T. E. 1993. Accurate methods for the statistics of surprise and coincidence. Computational Linguistics 19(1):61–74.

Evert, Stefan and Brigitte Krenn. 2005. Using small random samples for the manual evaluation of statistical association measures. Computer Speech and Language 19:450–466.

Fazly, Afsaneh. 2007. Automatic Acquisition of Lexical Knowledge about Multiword Predicates. Ph.D. thesis, University of Toronto.

Fazly, Afsaneh and Suzanne Stevenson. 2007. Automatic Acquisition of Knowledge about Multiword Predicates. In Proceedings of PACLIC 19, the 19th Asia-Pacific Conference on Language, Information and Computation.

Finlayson, Mark Alan and Nidhi Kulkarni. 2011. Detecting Multi-words improves Word Sense Disambiguation. In Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011).

Frank, Anette, Tracy Holloway King, Jonas Kuhn, and John T. Maxwell III. 1998. Optimality Theory Style Constraint Ranking in Large-scale LFG Grammars. In Proceedings of the LFG98 Conference. CSLI Publications.

Frank, Robert. 2002. Phrase Structure Composition and Syntactic Dependencies. MIT Press, Cambridge.

Grefenstette, Gregory and Simone Teufel. 1995. Corpus-based method for automatic identification of support verbs for nominalization. In Proceedings of the 7th Meeting of the European Chapter of the Association for Computational Linguistics (EACL'95).

Grimshaw, Jane and Armin Mester. 1988. Light verbs and theta-marking. Linguistic Inquiry 9(2):205–232.

Habash, Nizar, Bonnie Dorr, and David Traum. 2003. Hybrid natural language generation from lexical conceptual structures. Machine Translation 18:81–27.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1).

Han, Chung-hye and Owen Rambow. 2000. The Sino-Korean light verb construction and lexical argument structure. In Proceedings of the Fifth International Workshop on Tree-Adjoining Grammars and Related Formalisms, TAG+5.

Han, Chung-hye, Juntae Yoon, Nari Kim, and Martha Palmer. 2000. A Feature based Lexicalized Tree Adjoining Grammar for Korean. Tech. rep., Institute for Research in Cognitive Science, University of Pennsylvania, http://www.cis.upenn.edu/~xtag/koreantag.

Hautli, Annette, Sebastian Sulger, and Miriam Butt. 2012. Adding an Annotation Layer to the Hindi/Urdu Treebank. Linguistic Issues in Language Technology 7.

Hoffmann, Thomas and Graeme Trousdale, eds. 2013. The Oxford Handbook of Construction Grammar. Oxford University Press.

Hook, Peter. 1974. The Compound Verb in Hindi. University of Michigan, Ann Arbor.

Huang, C.-T James. 1992. Complex Predicates in Control. In R. Larson, U. Lahiri, S. Iatridou, and J. Higginbotham, eds., Control and Grammar, pages 109–147. Kluwer Academic Publishers, Dordrecht.

Husain, Samar. 2009. Dependency Parsers for Indian languages. In Proceedings of the ICON 2009 Tools Contest: Indian Language Dependency Parsing.

Husain, Samar, 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.

Hwang, Jena D., 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 held in conjunction with ACL-2010.

Jespersen, Otto. 1965. A Modern English Grammar on Historical Principles, Part VI, Morphology. George Allen and Unwin Ltd.

Joshi, Aravind and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenburg and A. Salomaa, eds., Handbook of Formal Languages, vol. 3, pages 69–124. Springer.

Kachru, Yamuna. 1982. Conjunct verbs in Hindi and Persian. South Asia Review 6(3):117–126.

Kachru, Yamuna. 2006. Hindi. John Benjamins.

Kallmeyer, Laura and Rainer Osswald. 2013. Syntax-driven semantic frame composition in Lexicalized Tree Adjoining Grammars. Journal of Language Modelling 1(2):267–330.

Kearns, Kate. 1988. Light verbs in English. Manuscript, MIT (revised 2002).

Kilgarriff, Adam, Siva Reddy, Jan Pomikálek, and Avinesh PVS. 2010. A Corpus Factory for Many Languages. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)eventh International Conference on Language Resources and Evaluation (LREC'10).

Kipper, Karin, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English vebs. Language Resources and Evaluation 42(1):21–40.

Krenn, Brigitte. 2000. The usual suspects: data-oriented models for the identification and representation of lexical collocations. Ph.D. thesis, Saarland University.

Kulkarni, Amba. 2011. Agreement in Hindi Conjunct Verbs. In Proceedings of ICON-2011: 9th International Conference on Natural Language Processing.

Kunchukuttan, Anoop and Om P. Damani. 2008. A System for Compound Noun Multiword Expression Extraction for Hindi. In Proceedings of ICON-2008: 6th International Conference on Natural Language Processing.

Levin, Beth. 1993. English Verb Classes and Alternations, A Preliminary Investigation. University of Chicago Press.

MacQueen, James B. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pages 281–297. University of California Press.

Malik, Muhammad Kamran, Tafseer Ahmed, Sebastian Sulger, Tina Bögel, Atif Gulzar, Ghulam Raza, Sarmad Hussain, and Miriam Butt. 2010. Transliterating Urdu for a Broad-Coverage Urdu/Hindi LFG Grammar. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010).

Manning, Christopher D. 1993. Analyzing the Verbal Noun: Internal and External Constraints. Japanese/Korean Linguistics 3:236–253.

Manning, Christopher D and Hinrich Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge.

Masica, Colin. 1976. Defining a linguistic area: South Asia. University of Chicago Press.

Mel'čuk, Igor A. 1988. Dependency Syntax: Theory and Practice. State University Press of New York.

Mohanan, Tara. 1994. Argument Structure in Hindi. CSLI Publications, Stanford.

Mohanan, Tara. 1997. Multidimensionality of representation- NV complex predicates in Hindi. In A. Alsina, J. Bresnan, and P. Sells, eds., Complex Predicates. CSLI Publications, Stanford.

Mukerjee, Amitabha, Ankit Soni, and Achla M Raina. 2006. Detecting Complex Predicates in Hindi using POS projection. In Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, pages 28–35.

Müller, Stefan. 2010. Persian Complex Predicates and the Limits of Inheritance-Based Analyses. Journal of Linguistics 46(3):601–655.

Narayan, Dipak, Debasri Chakrabarti, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. Experiences in Building the Indo WordNet- A WordNet for Hindi. In First International Conference on Global WordNet, Mysore, India.

North, Ryan. 2005. Computational Measures of the Acceptability of Light Verb Constructions. Ph.D. thesis, University of Toronto.

Pal, Santanu, Tanmoy Chakraborty, and Sivaji Bandopadhyay. 2011. Handling Multi-word expressions in Phrase-based Statisical Machine Translation. In Proceedings of the Workshop on Multiword Expressions (MWE 2011), 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011).

Palmer, Martha, Olga Babko-Malaya, Ann Bies, Mona Diab, Mohammed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A pilot Arabic PropBank. In Proceedings of the 6th International Language Resources and Evaluation.

Palmer, Martha, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure. In Proceedings of ICON-2009: 7th International Conference on Natural Language Processing. Hyderabad.

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

Pedersen, Ted, Satanjeev Bannerjee, Bridget T. McInnes, Saiyam Kohli, Mahesh Joshi, and Ying Liu. 2011. The n-gram statistics package- a flexible tool for identifying ngrams, collocations and word associations. In Proceedings of the Workshop on Multiword Expressions (MWE 2011), 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011).

Prince, Alan and Paul Smolensky. 2004. Optimality Theory: Constraint Interaction in Generative Grammar. Blackwell Publishing.

Rambow, Owen and Young-Suk Lee. 2007. Word Order Variation and Tree-Adjoining Grammar. Computational Intelligence .

Ramisch, Carlos, Paolo Schreiner, Marco Idiart, and Aline Villavicencio. 2008. An evaluation of methods for the extraction of multiword expressions. In Proceedings of the LREC Workshop towards a Shared Task for Multi Word Expressions.

Ratnaparkhi, A. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, MIT.

Reddy, Siva and Serge Sharoff. 2011. Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources. In Proceedings of the Fifth International Workshop On Cross Lingual Information Access, pages 11–19. Chiang Mai, Thailand: Asian Federation of Natural Language Processing.

Sadeghi, Ali Ashraf. 1993. On denominative verbs in Persian. In Farsi Language and the Language of Science, pages 236–246. Tehran: University Press.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword Expressions: A Pain in the neck for NLP. In Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'02), pages 1–15.

Schulte im Walde, Sabine and Chris Brew. 2002. Inducing German Semantic Verb Classes from Purely Syntactic Subcategorisation Information. In Proceedings of the 4th Annual Meeting of the Association for Computational Linguistics.

Sinha, R. Mahesh K. 2009. Mining Complex Predicates in Hindi Using a Parallel Hindi-English Corpus. In Proceedings of Workshop on Multiword Expressions, ACL-IJCNLP 2009, pages 40–46.

Stevenson, Suzanne, Afsaneh Fazly, and Ryan North. 2004. Statistical measures of the semi-productivity of light verb constructions. In Proceedings of the 2nd ACL Workshop on Multiword Expressions: Integrating Processing.

Sulger, Sebastian and Ashwini Vaidya. 2014. Towards Identifying Hindi/Urdu Noun Templates in Support of a Large-Scale lfg Grammar. In Proceedings of the Fifth Workshop on South and Southeast Asian Natural Language Processing at COLING 2014.

Swier, Robert S. and Suzanne Stevenson. 2004. Unsupervised Semantic Role Labelling. In Proceedings of the 2004 conference on Empirical Methods in Natural Language Processing.

Tan, Yee Fan, Min-Yen Kan, and Hang Cui. 2006. Extending corpus based identification of light verb constructions using a supervised learning framework. In Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context.

Taslimipoor, Shiva, Afsaneh Fazly, and Ali Hamzeh. 2012. Using Noun Similarity to Adapt an Acceptability Measure for Persian Light Verb Constructions. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, eds., Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). Istanbul, Turkey: European Language Resources Association (ELRA).

Tu, Yuancheng and Dan Roth. 2011. Learning english light verb constructions: Contextual or statistical. In Proceedings of the Workshop on Multiword Expressions (MWE 2011), 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011).

Vaidya, Ashwini, Jinho D. Choi, Martha Palmer, and Bhuvana Narasimhan. 2011. Analysis of the Hindi proposition bank using dependency structure. In Proceedings of the 5th Linguistic Annotation Workshop - LAW V '11.

Vaidya, Ashwini, Jinho D. Choi, Martha Palmer, and Bhuvana Narasimhan. 2012. Empty Argument Insertion in the Hindi PropBank. In Proceedings of the Eighth International Conference on Language Resources and Evaluation - LREC-12, Istanbul.

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

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

Vaidya, Ashwini, Martha Palmer, and Bhuvana Narasimhan. 2013. Semantic roles for nominal predicates: Building a lexical resource. In Proceedings of the 9th Workshop on Multi-word Expressions, NAACL-13.

Vaidya, Ashwini, Owen Rambow, and Martha Palmer. 2014. Light verb constructions with 'do' and 'be' in Hindi: A TAG Analysis. In Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing at COLING 2014.

Van de Cruys, Tim. 2006. Semantic Clustering in Dutch. In Proceedings of the Sixteenth Computational Linguistics in Netherlands (CLIN), pages 17–32.

Vijay-Shanker, K. and Aravind Joshi. 1988. Feature structure based Tree Adjoining Grammars. In Proceedings of COLING 1988.

Vincze, Veronika, István Nagy T, and Gabór Berend. 2011. Detecting noun compounds and light verb constructions: a contrastive study. In Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011).

Xia, Fei. 1999. Extracting Tree Adjoining Grammars from Bracketed Corpora. In Proc. of 5th Natural Language Processing Pacific Rim Symposium.

XTAG-Group, The. 2001. A Lexicalized Tree Adjoining Grammar for English. Tech. rep., IRCS, University of Pennsylvania.

Xue, Nianwen and Seth Kulick. 2003. Automatic Predicate Argument Structure Analysis of the Penn Chinese Treebank. In Proceedings of Machine Translation Summit IX (MTIX 2003)..

Xue, Nianwen 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.

Appendix A

Ontological categories from Hindi WordNet used for noun classification

(1) Abstract,Inanimate,Noun

(2) Action,Abstract,Inanimate,Noun

(3) Adverb

(4) Anatomical,Object,Inanimate,Noun

(5) Art,Abstract,Inanimate,Noun

(6) Artifact,Object,Inanimate,Noun

(7) Cognition,Abstract,Inanimate,Noun

(8) Communication,Action,Abstract,Inanimate,Noun

(9) Edible,Object,Inanimate,Noun

(10) Event,Inanimate,Noun

(11) Fatal_Event,Event,Inanimate,Noun

(12) Group,Noun

(13) Mammal,Fauna,Animate,Noun

(14) Mental_State,State,Noun

(15) Natural_Event,Event,Inanimate,Noun

(16) Natural_Object,Object,Inanimate,Noun

(17) Object,Inanimate,Noun

(18) Occupation,Action,Abstract,Inanimate,Noun

(19) Part_of,Noun

(20) Perception,Abstract,Inanimate,Noun

(21) Period,Time,Abstract,Inanimate,Noun

(22) Person,Mammal,Fauna,Animate,Noun

(23) Physical,Action,Abstract,Inanimate,Noun

(24) Physical_Place,Place,Inanimate,Noun

(25) Physical_Process,Process,Noun

(26) Physiological_State,State,Noun

(27) Place,Adverb

(28) Place,Inanimate,Noun

(29) Planned$_E$vent,Event,Inanimate,Noun

(30) Process,Noun

(31) Psychological_Feature,Abstract,Inanimate,Noun

(32) Qualitative,Descriptive,Adjective

(33) Quality,Abstract,Inanimate,Noun

(34) Quantitative,Descriptive,Adjective

(35) Relational,Adjective

(36) STY,Art,Abstract,Inanimate,Noun

(37) Shape,Descriptive,Adjective

(38) Social,Action,Abstract,Inanimate,Noun

(39) State,Noun

(40) Stative,Descriptive,Adjective

(41) Time,Abstract,Inanimate,Noun

(42) Verb_of_Action,Verb

(43) Action,Descriptive,Adjective

(44) Concept,Abstract,Inanimate,Noun

(45) Information,Abstract,Inanimate,Noun

(46) Physical_State,State,Noun

(47) Possession,Abstract,Inanimate,Noun

(48) Property,Abstract,Inanimate,Noun

Appendix B

'Verb-centric' analysis in TAG

In this section, we explore a verb-centric analysis in the LTAG framework. This analysis, captures a more equitable distribution of arguments within the NVC (c.f. Figure 5.6). According to this view, both noun and light verb contribute their arguments to the LVC. In this analysis, the light verb is depicted as an **initial** tree, and the nominal as an **auxiliary** tree. The nominal adjoins into the light verb's tree, in a mirror-image of the analysis presented before.The light verb has an incomplete argument structure, which requires adjunction of the nominal. A nominal that alternates between two light verbs will have a single tree that will adjoin into the appropriate light verb's initial tree.

For convenience, we repeat the two examples for the NVC *tareef kar* in 65 and 66. In the verb-centric analysis, the light verb *kar* 'do' will have its own agentive argument, whereas the light verb *ho* 'be' will not have arguments of its own. The nominal *tareef* 'praise' will have its own argument 'pustak'. (Note that the genitive *kii* and inflected form of *kar* 'do' as *kii* (fem.sg) have no relation with each other)

(65) *pustak-kii tareef huii.*
book.F-Gen praise.F be.Perf.F
'The book got praised'

(66) *logon-ne pustak-kii tareef kii.*
people.M-Erg book.F-Gen praise.F do.Perf.F.Pl
'People praised the book '

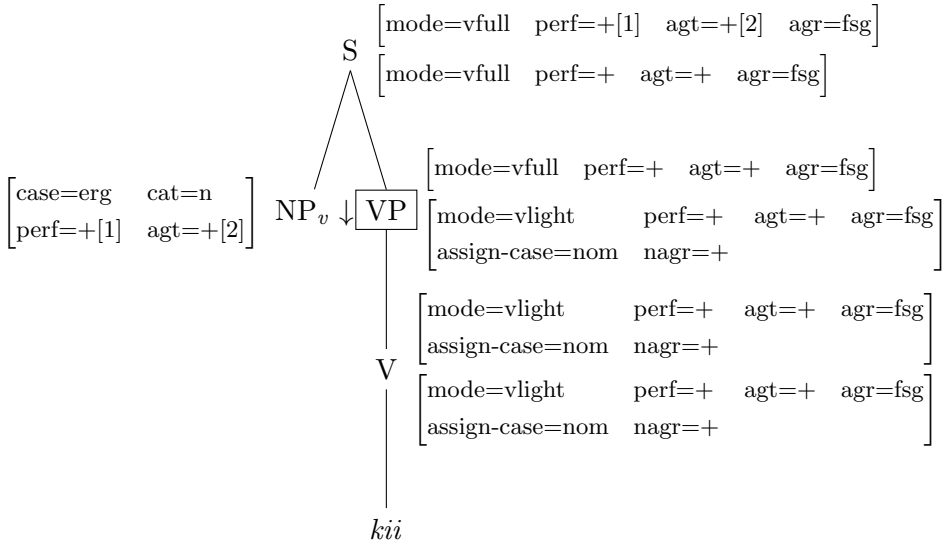

Figure B.1: The elementary tree for the light verb *kar* 'do', which is an initial tree

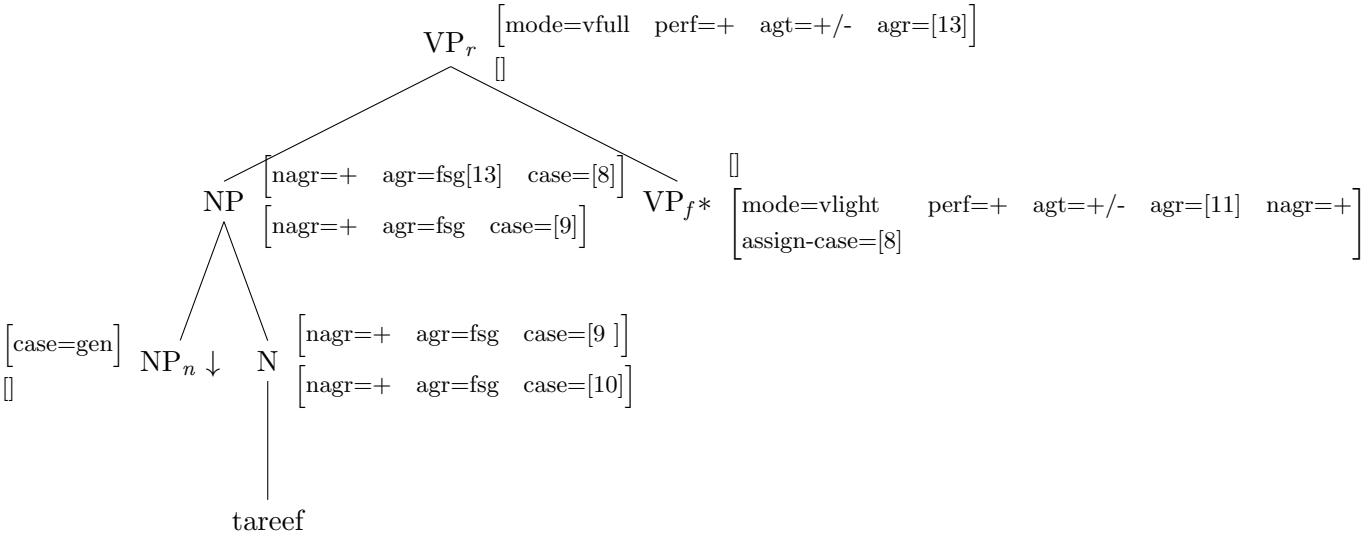

Figure B.2: The elementary tree for the nominal *tareef*, which is an auxiliary tree

Figure B.2 and Figure B.1 depart from the original TAG analysis in a few ways - first the light verb is an initial tree which forms a syntactic predication structure. However, as compared to other 'full' verbs, the light verb requires another predicating element to complete its argument structure. The feature `MODE` captures this incompleteness. We assume that the S node is always specified for `MODE=vfull` but at VP, the bottom feature structure is `MODE=vlight` (Figure B.1). For the light verb *ho* 'be', the analysis will be similar in that there will be no agentive argument supplied by *ho* 'be'. The initial tree for *ho* will also force adjunction at its VP node.

It might seem plausible at first to propose an analysis of substitution into the light verb's tree at NP_n in figure B.1. However, this analysis is not preferable as we will not be able to differentiate between light and 'full' *kar* 'do' (see also Figure **??**). The feature clash that forces adjunction at the VP node tells us that a second predicate is required to complete the argument structure (and not only an argument NP).

The feature clash will allow the tree for the nominal *tareef* 'praise' in Figure B.2 to adjoin at VP. This tree is an auxiliary tree, with its root and foot nodes as VP. It can combine with a tree having either positive or negative features for `AGT`. For examples like 65 and 66, there will be a single elementary tree representing *praise*. The `CASE` feature at the NP node in Figure B.2 consists of variables that are not yet specified for their value. They will get `CASE` as nominative post adjunction into the verb's initial tree. Other features in the tree are familiar from the noun-centric analysis: `NAGR=+` as the light verb shows agreement with *praise*, `PERF` captures the fact that the verb is perfective and the subject has ergative case.

Post adjunction, the feature clash at VP on the light verb's tree will be resolved and the appropriate `AGR` features will be unified. This tree is shown in Figure B.3. At VP_f, the auxiliary tree's bottom feature structure (Fig B.2) unifies with the initial tree's bottom feature structure (Fig B.1). Similarly, the top feature structure of the auxiliary tree at VP_r unifies with the top feature structure of the initial tree. After this step, there are no remaining feature clashes in the tree. Therefore, a final step of unification of features at each node will take place and this is shown in Figure B.4. The nominal *praise* gets nominative case from the light verb and the arguments

logon-ne and *pustak-kii* will substitute into NP_v and NP_n respectively.

The light verb *ho* 'be"s initial tree will not have arguments of its own. At its VP node, we will model a similar feature clash between the `light` and `full` feature values, which will force the adjunction of *praise* into the tree. Note, that the design of the elementary tree for *ho* may not always appear without arguments. If, for instance we wish to model the argument composition for an example such as 41, then *ho* will appear with a experiencer-case marked argument. We will prevent adjunction of *praise* into such a tree by the addition of a feature denoting the semantic type of noun. E.g. we know that only psychological nouns will result in an experiencer case marked subject with *ho* (but not abstract action nouns such as *praise*).

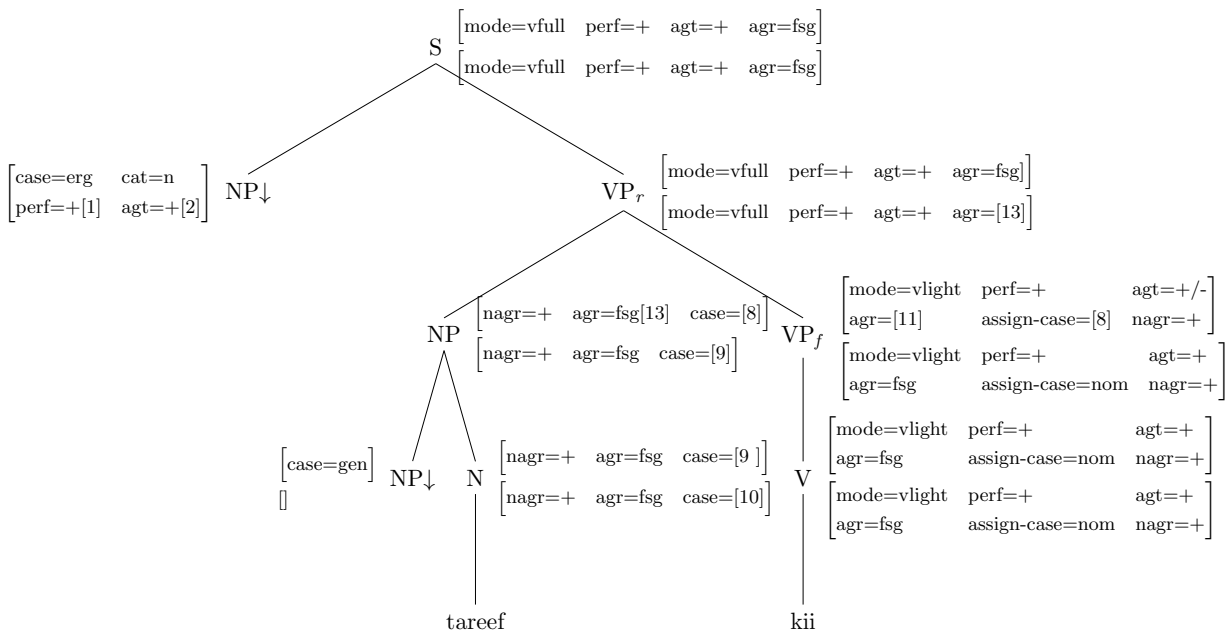

Figure B.3: After adjunction of the nominal into the light verb's elementary tree, we get a composed structure for *tareef kii*

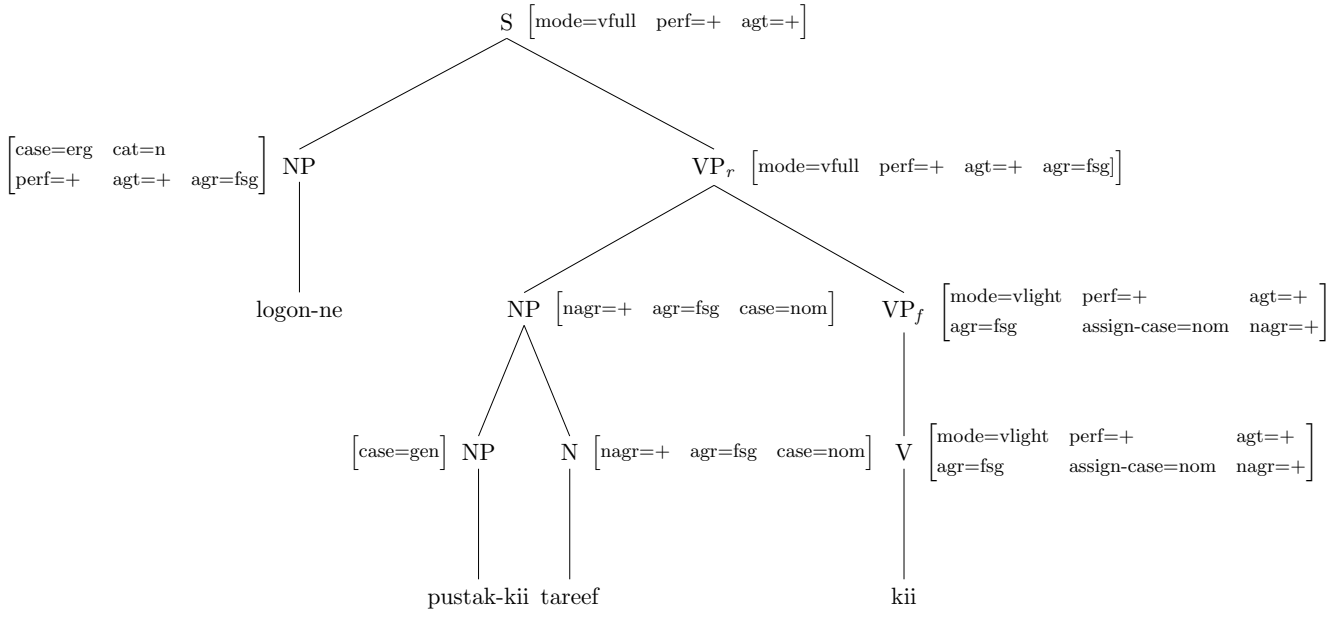

Figure B.4: After the final step of top and bottom unification of features at each node, we get the final composed tree shown above.