Linguistic **I**ssues in **L**anguage **T**echnology – **LiLT**
Submitted, January 2012

Adding an Annotation Layer to the Hindi/Urdu Treebank

Annette Hautli
Sebastian Sulger
Miriam Butt

Published by CSLI Publications

LiLT volume 7, issue 3 January 2012

Adding an Annotation Layer to the Hindi/Urdu Treebank

Annette Hautli, *Universität Konstanz* Sebastian Sulger, *Universität Konstanz* Miriam Butt, *Universität Konstanz*

Abstract

This paper proposes an additional layer of annotation for the recently established Hindi/Urdu Treebank. Despite the fact that the treebank already features a number of annotation layers such as phrase structure, dependency relations and predicate-argument structure, we see potential for the inclusion of a dependency layer generated from Lexical-Functional Grammar (lfg) f-structures with relations that we believe are crucial for a deep analysis of Urdu/Hindi. The suggestions are based on theoretical and computational investigations into Urdu/Hindi in the context of the Urdu ParGram grammar, through which we can automatically create the additional annotation layer.

LiLT Volume 7, Issue 3, January 2012.
Adding an Annotation Layer to the Hindi/Urdu Treebank.

1 Introduction

Many statistical natural language processing (NLP) applications rely on treebanks where syntactic and semantic patterns of language are annotated in order to provide a relatively comprehensive sample of linguistic constructions with a theoretically well-founded representation of these constructions. Having a variety of annotation levels available and in particular information on the semantic structure, machine learning can reliably predict linguistic patterns. A multi-layered treebank is of particular importance if few other resources for a language exist.

The Hindi/Urdu Treebank (henceforth HUTB) (Palmer et al., 2007, Bhatt et al., 2009) is a novel attempt to create a multi-layered treebank for Indo-Aryan languages; it features different annotation levels, namely a phrase structure annotation inspired by the Chomskyan approach to syntax (Chomsky, 1981, 1995), a level of dependency annotation following the Computational Pāṇinian Grammar (Bharati et al., 1995, Begum et al., 2008) as well as the marking of predicate-argument structure in the PropBank style (Palmer et al., 2005). The dependency annotation mainly expresses verb-centric relations as developed by Pāṇini, i.e. the relation of arguments with respect to a given verb. These relations can be divided into *kaṛaka*- (e.g., agent, theme, etc.), non-*kaṛaka*-labels and *modifier*-labels.

The Urdu ParGram grammar (Butt and King, 2007, Bögel et al., 2007, 2009), another resource for the otherwise rather sparsely resourced language of Urdu, is a computational grammar based on the Lexical-Functional Grammar formalism. Hindi and Urdu are mostly parallel with respect to syntax and semantics, some differences exist in the domain of vocabulary. Due to the close relationship between Hindi and Urdu, the investigations and in particular the linguistic analyses and their potential representation by dependencies are generally valid across both languages.

Given this, a natural research question is: Can the HUTB be augmented using the analyses produced by the Urdu ParGram grammar? Our answer to this question is yes; by adding to the verb-centric approach of the HUTB, we propose to add another level of annotation on a separate layer, providing additional detail and usability to the HUTB.

In order to introduce the additional annotation layer, we proceed as follows: Section 2 introduces the various ingredients that are involved in providing an additional layer to the HUTB. The proposed layer of annotation is presented in Section 3, in particular we will discuss the dependency annotation of linguistic phenomena such as modality and tense/aspect (in 3.1 and 3.2, respectively). This is followed by our anno-

tation of multiword expressions in 3.3 and the management of syntactic ambiguity that is elaborated on in Section 3.4. Section 4 concludes the paper.

2 Ingredients

2.1 The Hindi/Urdu Treebank

The Hindi/Urdu Treebank (Palmer et al., 2007, Bhatt et al., 2009) is a *multi-layered* treebank in that it includes three levels of annotation, namely two syntactic levels and one lexical-semantic level. One syntactic level is annotated with phrase structure inspired by the Chomskyan approach to syntax (Chomsky, 1981, 1995) and assumes a binary branching representation. The other level is a dependency structure which follows the Computational Pāṇinian Grammar (CPG) (Bharati et al., 1995, Begum et al., 2008).

Pāṇini developed the earliest known work on descriptive linguistics around 400 BCE, in which an extremely elegant, compact and efficient system of rules encode a grammar for Sanskrit. The morphosyntactic and lexical semantic aspects of the grammar are essentially equivalent to today's dependency grammars in that dependencies between a syntactic head and its arguments and adjuncts are encoded. In particular, Pāṇini developed a system of *kaṛaka* relations (essentially equivalent to today's thematic roles; Kiparsky and Staal 1969) which he placed into correspondence with case marking and differences in lexical semantic interpretation (see Butt (2006) for a short discussion and pointers to relevant references). The basic *kaṛaka* relations first posited by Pāṇini are integrated and used as part of the CPG layer of annotation in the HUTB. Although the *kaṛaka* relations are lexical semantic in nature, like Pāṇini, the CPG goes beyond just these lexical semantic relations and provides an essentially (morpho)syntactic dependency annotation.

At another level of annotation, a purely lexical semantic one, the dependency relations are encoded according to the English PropBank (Palmer et al., 2005), where semantic roles such Arg0, Arg1 etc. are assigned to the arguments of the verb. These PropBank roles can be mapped onto Pāṇini's *kaṛaka* roles.

The treebank is also *multi-representational* in that the two syntactic levels of annotation encode linguistic phenomena differently, e.g. long-distance dependencies are encoded in the dependency relation whereas they are not present in the Chomskyan-style phrase structures.

See (1) for a Hindi treebank example[1,2] with its syntactic annota-

[1] File `fullnews_id_2489467_date_30_5_2004.dat`, sentence ID 6

[2] The Roman transliteration scheme used in this paper for both Hindi, written

tions in Figure 1 (with phrase-structure annotations in bold face and dependency relations in italics).

(1) दूतावास ऐध्कारियों ने उसे अच्छी सेह्त में पाया

dUtAvAs adHikAriyOn=nE usE accHI
embassy.Masc.Sg officer.Masc.Pl=Erg he=Acc good.Fem.Sg

sEhat=mEN pAyA
health.Fem.Sg=Loc find.Perf.Masc.Sg
'Embassy officers found him in a healthy condition.'

```
(( NP                          <fs drel='k1:VGF' name='NP'>
        XC      दूतावास
        NN      ऐध्कारियों
        PSP     ने
))
(( NP                          <fs drel='k2:VGF' name='NP2'>
        PRP     उसे
))
(( NP                          <fs drel='k7:VGF' name='NP3'>
        JJ      अच्छी
        NN      सेह्त
        PSP     में
))
(( VGF
        VM      पाया
        SYM     .
))
```

FIGURE 1 Sample sentence from the Hindi/Urdu Treebank

The HUTB has about 30 dependency relations, the ones starting with 'k' are Pāṇinian thematic roles, the *kaṛaka* relations (Kiparsky and Staal, 1969), and are assigned to the arguments of a verb (VGF). Figure 1 also encodes the phrase-structure of (1), whereby the preterminal nodes assign a part of speech to each lexical item (e.g. XC, NN, PSP). These parts of speech are grouped into constituents as part of the sentence analysis. The dependencies are attached to the constituent level, marked with 'drel'. *k1* is the agent of an action (here, *dUtAvAs adHikAriyOn* 'embassy officers'), whereas *k2* is the object or patient of a

in Devanagari script, and Urdu, written in a version of Arabic script, is described in Malik et al. (2010).

verb (here, *usE* 'him'). *k7* is an argument which supports *k1* or *k2* in a locative or temporal manner, in this case the 'in good health' has been classified as such a relation. The PropBank frame for the verb *pA* 'find' in (1) is given in Table 1.

पा 'to find'		
Arg0	agent	दूतावास ऐध्कारियों 'embassy officers'
Arg1	patient (theme)	उसे 'him'
ArgM-MNR	modifier (manner)	अच्छी सेह्त 'good health'

TABLE 1 PropBank frame for पा 'to find' in (1)

2.2 Lexical-Functional Grammar

Lexical-Functional Grammar (LFG) (Bresnan and Kaplan, 1982, Dalrymple, 2001) is a constraint-based grammar formalism. LFG assigns two levels of syntactic description to every sentence of a language. Phrase structure configurations are represented in a *constituent structure* (or 'c-structure'), indicating the surface arrangement of words and phrases in the sentence as well as constituency and hierarchical relations among the constituents. Grammatical relations are represented explicitly at the other level of description, the *functional structure* (or 'f-structure'), which encodes traditional syntactic notions such as subject, object, complement and adjunct in the form of an attribute-value matrix.

The Urdu ParGram grammar (Butt and King, 2007, Bögel et al., 2007, 2009) is part of an international research program called ParGram (Parallel Grammars) that is aimed at developing parallel syntactic analyses for different languages within the framework of LFG (Butt et al., 1999, 2002), using the development platform XLE (Crouch et al., 2011). 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. See Figure 2 for the f-structure of (1), produced by the Urdu ParGram grammar.

For the purpose of this paper, we restrict ourselves to the f-structure representation of (1), as only functional information (and not constituency information as in the c-structure) is considered for the dependency triples that are generated by the Urdu ParGram grammar. The nature of these triples is discussed in the following section.

FIGURE 2 F-structure for (1)

2.3 Dependency triples

LFG explicitly encodes dependency information by means of the f-structure. These dependencies can be reformulated as triples, where the triple `subj(pA, dUtAvAs adHikAriyOn)` expresses that the `subj` of the verb `pA` 'find' is the predicate `dUtAvAs adHikAriyOn` 'embassy officers'. Reformulating f-structures into these dependency triples has also been done for parts of the Wall Street Journal section of PennTreebank, resulting in PARC700 (King et al., 2003), a gold dependency bank generated by the English ParGram grammar (Bobrow et al., 2007).

XLE allows the creation of triples out of f-structure relations via an internal process which is flexible enough so that features can also be deleted; i.e., the choice of the set of triples features is up to the designer of the dependency bank. See below for an exemplary set of triples, where only the grammatical function triples of the f-structure in Figure 2 have been selected.

(2)
```
pred(root,pA)
subj(pA,dUtAvAs adHikAriyOn)
obj(pA,vuh)
adjunct(pA,sEhat)
adjunct(sEhat,accH)
```

Using these triples, we propose to add an additional layer of annotation to the HUTB which is not restricted to just verb-related dependencies such as *kaṛaka* and modifier roles, but contains additional information as detailed below in the next section.

3 An additional dependency annotation for the HUTB

We propose to use a triples dependency layer as an addition to the HUTB that is partly different from the dependency layer already available for it. Instead of following the model of Pāṇinian *kaṛaka* relations, we propose to generate a set of triples that is generated by f-structure relations of the Urdu ParGram grammar. However, the triples themselves are in principle theory-independent.

As to the choice of the set of dependency triples for the HUTB extension, we follow the proposal made for the PARC700 dependency bank in that we reduce the highly-articulated XLE f-structures (as in Figure 2) to attributes with `PRED` values (all grammatical functions), attributes with positive (+) values and an additional set of features which is partly parallel to the PARC700 feature set and has proven useful in statistical experiments (Hautli et al., 2010). See Table 2 for a list.

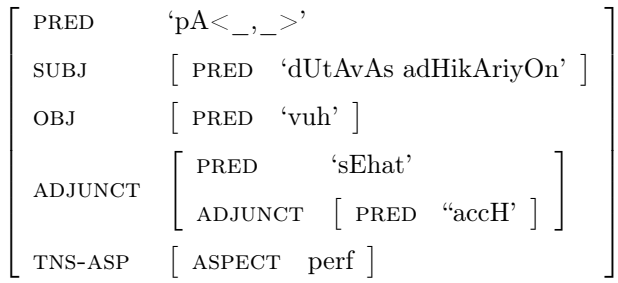
Grammatical function labels			
subj	subject	obj	object
obl	oblique	comp	compl. clause
xcomp	open complement	predlink	copula constr.
adjunct	adjunct phrase	conj	conjunction
topic	topic phrase	focus	focus phrase
poss	possessive phrase	mod	modifier clause

Other feature labels			
feature	**value**	**feature**	**value**
address	e.g. rude, familiar	adv-type	e.g. loc, sadv
adjunct-type	e.g. loc	aspect	e.g. prog
case	e.g. erg, acc, dat	causative	direct, indirect
coord-form	e.g. and, because	deixis	e.g. proximal
gend	masc, fem	mood	e.g. imperative
mod-type	e.g. ezafe	num	sg, pl
modality	e.g. must, can	tense	e.g. past
number-type	card, ord	passive	+
pron-type	e.g. pers, rel	proper-type	e.g. location,
vtype	e.g. main, copular		name

TABLE 2 Labels for an additional dependency annotation

For illustration purposes, the f-structure in Figure 2 can be reduced to the one in Figure 3. For the set of triples below it, we only keep the grammatical functions and one attribute-value pair that is flattened, namely [TNS-ASP [ASPECT perf]] is flattened to `aspect(pA,perf)`. Having the `tns-asp` f-structure is mainly an architectural decision of the ParGram community. The attribute-value pairs encoding the particular tense/aspect contained in the `tns-asp` feature are the ones which contain crucial information and which should be preserved in the triples. This yields the set of dependency triples at the bottom of Figure 3.

We would like to emphasize we do not as yet provide a complete dependency bank for the HUTB, but in this paper provide an argument that developing it would provide a useful extra resource. In the following, we will discuss some sample phenomena where we propose that the analyses of the Urdu ParGram grammar can generate important additional dependency information, thereby enhancing the usability of the HUTB for NLP applications.




```
pred(root,pA)
subj(pA,dUtAvAs adHikAriyOn)
obj(pA,vuh)
adjunct(pA,sEhat)
adjunct(sEhat,accH)
aspect(pA,perf)
```

FIGURE 3 Reduced f-structure and set of triples for (1)

3.1 Modality

Urdu/Hindi features only two modal verbs with a defective paradigm, while all other modality is expressed constructionally by combining the main verb with either *sak* 'can', *pa* 'find', *par* 'fall' and *ho* 'be' (Bhatt et al., 2011).

While the HUTB annotates modal constructions with the PropBank label `ARGM-Mod`, it does not encode information about the kind of modality expressed by the constructions. We therefore propose to include information on it via our dependency annotation. Due to the prevalence of constructional modality in Urdu/Hindi, this information is already encoded in the f-structure representation. See the sentence in (3), where the bare form of *kar* 'do' combines with an inflected form of *sak* 'can'.

(3) yAsIn vuh kar sakA
Yasin.Masc.Sg.Nom that.Sg.Nom do.Bare can-Perf.Masc.Sg
'Yasin could do that.'

Figure 4 shows the XLE f-structure representation for the example in (3). Figure 5 shows the reduced XLE f-structure, where the attribute-value pair [MODALITY can] is crucially retained to form a part of the set of triples.

```
"yAsIn vuh kar sakA"

[PRED    'sak<[22:kar]>[1:yAsIn]'
 SUBJ    [PRED  'yAsIn'
          NTYPE [NSEM [PROPER [PROPER-TYPE name]]
                 NSYN proper]
        1[CASE nom, GEND masc, NUM sg, PERS 3 ]]
 XCOMP   [PRED    'kar<[1:yAsIn], [19:vuh]>'
          SUBJ    [1:yAsIn]
          OBJ     [PRED  'vuh'
                   NTYPE [NSYN pronoun]
                 19[CASE nom, NUM sg, PERS 3, PRON-TYPE pers]]
        22[PASSIVE -]
 TNS-ASP [ASPECT perf, MOOD indicative]
51[CLAUSE-TYPE decl, MODALITY CAN, VTYPE main ]
```

FIGURE 4 F-structure for (3)

$$\begin{bmatrix} \text{PRED} & \text{`sak<\_,\_>'} \\ \text{SUBJ} & [\ \text{PRED}\ \ \text{`yAsIn'}\ ] \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{`kar'} \\ \text{SUBJ} & [\ \text{PRED}\ \ \text{`yAsIn'}\ ] \\ \text{OBJ} & [\ \text{PRED}\ \ \text{`vuh'}\ ] \end{bmatrix} \\ \text{MODALITY} & \text{CAN} \end{bmatrix}$$

```
pred(root,sak)
subj(sak,yAsIn)
xcomp(sak,kar)
subj(kar,yAsIn)
obj(kar,vuh)
modality(sak,can)
```

FIGURE 5 Reduced f-structure and set of triples for (3)

3.2 Tense, aspect and mood

Tense, aspect and mood is another area where we argue that our grammar can enhance the annotation in the HUTB. While Urdu/Hindi is a language with an elaborate system for expressing different aspectual and temporal notions, such as progressive, continuation, habitual, iterative, perfective and imperfective readings (Butt and Rizvi, 2010), the

HUTB does not differentiate between any of these. Aspectual auxiliaries receive the same part of speech tag as tense auxiliaries (`VAUX`); the nuances in the tense/aspect system are therefore not accommodated, and linguistic concepts such as *past tense* or *perfective aspect* are not annotated. An example from the HUTB is given in (4).[3] The XLE parse of the Urdu ParGram grammar for this sentence is shown in Figure 6.

(4) lAlU yAdav Ek sAtH dO mOrcON=par
Lalu Yadav one-time two rally.Masc.Pl.Obl=Loc

kAm kar rahA tHA
work do Prog.Masc.Sg be.Impf.Masc.Sg
'Lalu Yadav was working on two rallies at once.'

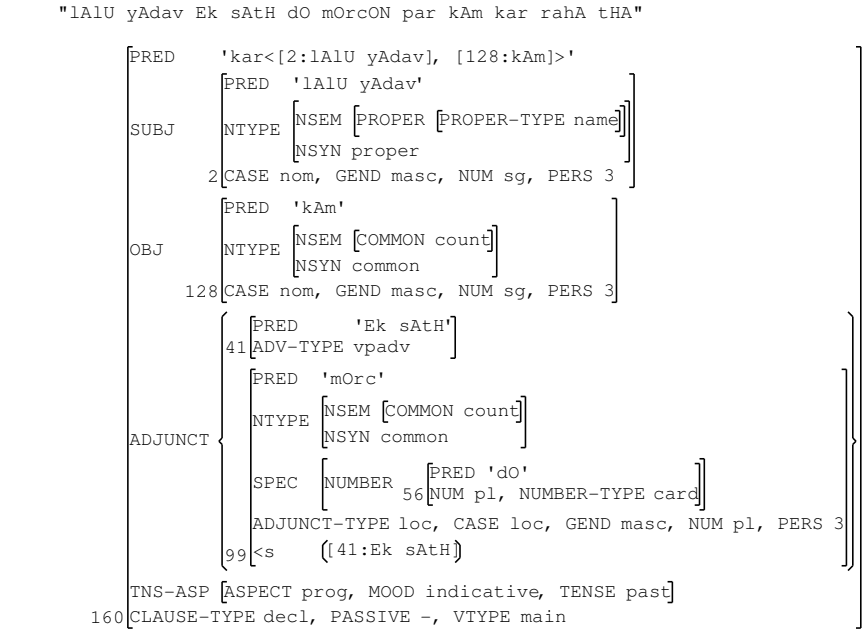
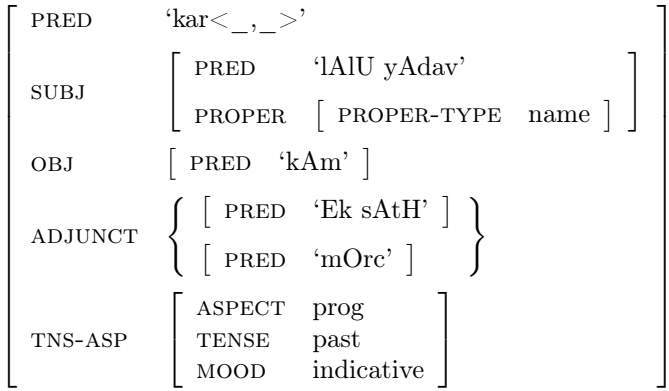


FIGURE 6 F-structure for (4)

Figure 7 shows the reduced f-structure, where ASPECT, TENSE and MOOD features are kept together with their respective values for annotation in the dependency triples for the HUTB. The figure also shows

[3]File `fullnews_id_2508874_date_8_6_2004.dat`, sentence ID 8; slightly adapted example.

the triples proposed for the annotation of multiword entities which is dealt with in Section 3.3.




```
pred(root,kar)
subj(kar,lAlU yAdav)
obj(kar,kAm)
adjunct(kar,Ek sAtH)
adjunct(kar,mOrc)
proper-type(lAlU yAdav,name)
aspect(kar,prog)
tense(kar,past)
mood(kar,indicative)
```

FIGURE 7 Reduced f-structure and set of triples for (4)

3.3 Multiword entities

Multiword entities (MWEs) are sequences of words that together form a single lexical entry. MWEs may be named entities (e.g., *New York*, *Golden Gate Bridge*, *George Bush*), verbal expressions (e.g., *kick the bucket*), or even quantifiers (e.g., *a lot of*). MWEs are often syntactically and semantically idiosyncratic in nature which makes their identification an important task for NLP applications. Instead of analyzing the parts separately, they need to be treated as a complex unit.

In the HUTB, MWEs are marked using the tag XC, where X is a variable indicating the type of compound of which the word in question is a part of; NNC is the tag used for common noun compounds, while NNPC is the tag used for proper noun compounds. Note that any further indication as to the nature of the MWE is missing from the annotation,

i.e., whether it represents a named entity, a phrase, an idiom, etc. is not reflected in any of the annotations. The distinction is merely between common nouns and proper nouns. Note also that the HUTB currently only annotates nominal MWEs, not adjectival, adverbial or verbal ones, which also appear in Indo-Aryan languages like Urdu/Hindi.

As part of the Urdu ParGram grammar, we have designed a MWE lexicon, complete with morphosyntactic information and information regarding their semantic class (Hautli and Sulger, 2011). Consider the example in (4) again, and the f-structure in Figure 6. As can be seen in the PRED of the SUBJ f-structure, lAlU yAdav (a person name referring to an Indian politician) is correctly analyzed as an MWE. Moreover, based on the tags provided by the MWE lexicon, the grammar identifies the string as a name, indicated by the feature [PROPER-TYPE name]. The reduced f-structure for (4) has been given in Figure 7, along with the triples relevant for the MWE annotation.

3.4 Ambiguity management

During the annotation process of the HUTB, any ambiguities present in the examples were resolved, both through manual annotation using the context of the sentence and through the application of heuristics. The choice between the two disambiguation techniques depended on the level of analysis and the structure in question. For most purposes, the fact that the HUTB is fully disambiguated might not be a problem, however for users interested in investigating ambiguities, it might instead be advantageous to have access to the available ambiguities.

The Urdu ParGram grammar can detect ambiguities in sentences automatically, independent of their level. XLE stores and displays ambiguities using packed f-structures (King et al., 2000). These packed f-structures can then be disambiguated or the packed ambiguities can be passed along to another module as one representation (e.g., easing the classic problem of PP-attachment in machine translation applications, where the ambiguity often exists in both languages). The triples for each reading can be extracted independently based on Prolog representations of their respective f-structures.

Consider the potentially ambiguous sentence in (5). Parsing the sentence and converting each of the two resulting f-structures into the triples format produces the set of triples as depicted in Figures 8 and 9.

Note that this example actually involves a complex predicate, in this case a morphological causative. Like most South Asian languages, Urdu/Hindi makes prolific use of both morphological and periphrastic complex predicates. In the Urdu ParGram grammar, these are all dealt

with via the restriction operator, which can apply either in conjunction with syntactic rules (periphrastic complex predicates) or in conjunction with sublexical rules (morphological complex predicates like the causative in (5)). See Butt et al. (2003), Butt and King (2006), Butt et al. (2009) for some discussion.

(5) nAdiyah yAsIn=kO tArA
Nadya.Fem.Sg Yassin.Masc.Sg=Acc Tara.Fem.Sg

dEkHAtI he
see.Caus.Impf.Fem.Sg be.Pres.3.Sg

Reading 1: 'Nadya shows Yassin Tara.'
Reading 2: 'Tara shows Yassin Nadya.'

$$\begin{bmatrix} \text{PRED} & \text{'cause<\_, dEkH<\_,\_> >'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'nAdiyah'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'tArA'} \end{bmatrix} \\ \text{OBJ-GO} & \begin{bmatrix} \text{PRED} & \text{'yAsIn'} \end{bmatrix} \end{bmatrix}$$

```
pred(root,cause_dEkH)
subj(cause_dEkH,nAdiyah)
obj(cause_dEkH,tArA)
obj-go(cause_dEkH,yAsIn)
```

FIGURE 8 Reduced f-structure and set of triples for reading 1 of (5)

$$\begin{bmatrix} \text{PRED} & \text{cause<\_, dEkH<\_,\_> >'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'tArA'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'nAdiyah'} \end{bmatrix} \\ \text{OBJ-GO} & \begin{bmatrix} \text{PRED} & \text{'yAsIn'} \end{bmatrix} \end{bmatrix}$$

```
pred(root,cause_dEkH)
subj(cause_dEkH,tArA)
obj(cause_dEkH,nAdiyah)
obj-go(cause_dEkH,yAsIn)
```

FIGURE 9 Reduced f-structure and set of triples for reading 2 of (5)

Basically, the Restriction Operator allows for the building of a complex value for the PRED of the main clause, as illustrated in Figures 8 and 9. For the purposes of this paper, we have here abstracted away from the dependency relations between the parts of the complex predicate (i.e., in the f-structures in Figures 8 and 9, the causative embeds the predicate 'see') and have just represented the complex predicate as one unit in the triples representations.[4]

The resulting sets of triples can be included in the additional dependency layer of the HUTB as in (6), enabling the user to keep track of any ambiguities in the treebank. If the users do not wish to make use of the ambiguities, they can by default fall back on either set of triples.

(6)
```
choice(
(pred(root,cause_dEkH)
subj(cause_dEkH,nAdiyah)
obj(cause_dEkH,tArA)
obj-go(cause_dEkH,yAsIn))
,
(pred(root,cause_dEkH)
subj(cause_dEkH,tArA)
obj(cause_dEkH,nAdiyah)
obj-go(cause_dEkH,yAsIn))
)
```

We also propose to manually indicate which of the analyses is given in the original HUTB annotation. Note that XLE provides the grammar writer with various disambiguation tools of its own, including stochastic extensions and constraints in the style of optimality theory. The Urdu ParGram grammar makes use of these techniques; in cases where there are optimal and less than optimal analyses for a single sentence, we propose to mark optimal ones in the triples choice space.

4 Summary and outlook

In this paper, we propose an additional layer of dependency annotation for the HUTB. We present theoretical reasons why this particular layer of annotation provides information that has not yet been captured in the HUTB. In particular, we can provide linguistically informed analy-

[4] Given that there are several different types of complex predicates in Urdu/Hindi and that this information may also be semantically relevant, we actually are working on developing a more complex triples representation, but a discussion of this would lead us too far afield here.

ses for phenomena such as modality, tense/aspect/mood and syntactic ambiguity, yielding a more detailed dependency structure. Moreover, multiword entities can be classified and tagged correctly which is an important aspect for every NLP task.

As a more general point, we would like to emphasize that the dependency triples in the proposed additional layer can be mapped onto the basic HUTB format. While the dependency triples look quite different from the dependency structure in Figure 1, their format is nevertheless the same, encoding the head, the dependent and the kind of dependency; it is rather that the triples encode a greater level of detail. Therefore, the format of the dependency triples from our additional annotation layer can be rewritten in a straightforward way to fit in with the other layers of the HUTB.

The ultimate target is to provide a dependency annotation layer for the HUTB that is similar in style and size to the PARC700 dependency bank, which has proven to be highly effective when independently evaluating parsers for a language. In current and on-going work we are randomly selecting sentences from the HUTB and automatically adding our additional annotation layer while manually inspecting analyses.

The additional bonus of a dependency bank on the basis of HUTB is that parsers which are either mainly oriented towards Hindi or mainly oriented towards Urdu can be evaluated against one single gold standard. We therefore anticipate that our additional annotation layer can generally improve the training of linguistically informed parsers for Hindi and Urdu.

Acknowledgments

We gratefully acknowledge comments from two anonymous reviewers and we would like to thank the HUTB team, especially Rajesh Bhatt, for sharing the treebank data with us and educating us on particular aspects of the treebank. We also gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG), whose funding made this work and this paper possible.

References

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 of Natural Language Processing (IJCNLP)*. Hyderabad, India.

Bharati, Akshar, Vineet Chaitanya, and Rajeev Sangal. 1995. *Natural Language Processing — A Paninian Perspective*. Prentice Hall of India.

Bhatt, Rajesh, Tina Bögel, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2011. Urdu/Hindi Modals. In M. Butt and T. H. King, eds., *Proceedings of the LFG11 Conference*. Hong Kong.

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 *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Suntec, Singapore: Association for Computational Linguistics.

Bobrow, Daniel G., Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen. 2007. PARC's Bridge and Question Answering System. In *Grammar Engineering Across Frameworks*, pages 46–66. CSLI Publications.

Bögel, Tina, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2007. Developing a Finite-State Morphological Analyzer for Urdu and Hindi: Some Issues. In *Proceedings of FSMNLP07, Potsdam, Germany*.

Bögel, Tina, Miriam Butt, Annette Hautli, and Sebastian Sulger. 2009. Urdu and the Modular Architecture of ParGram. In *Proceedings of the Conference on Language and Technology 2009 (CLT09)*.

Bresnan, Joan and Ronald M. Kaplan. 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

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

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of COLING2002, Workshop on Grammar Engineering and Evaluation*, pages 1–7. Taipei, Taiwan.

Butt, Miriam and Tracy Holloway King. 2006. Restriction for morphological valency alternations: The Urdu causative. In M. Butt, M. Dalrymple, and T. H. King, eds., *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, pages 235–258. Stanford: CSLI Publications.

Butt, Miriam and Tracy Holloway King. 2007. Urdu in a Parallel Grammar Development Environment. *Language Resources and Evaluation* 41(2):191–207.

Butt, Miriam, Tracy H. King, and John T. Maxwell III. 2003. Complex Predicates via Restriction. In M. Butt and T. H. King, eds., *Proceedings of the LFG03 Conference*, pages 92–104. Stanford: CSLI Publications.

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

Butt, Miriam, Tracy Holloway King, and Gillian Ramchand. 2009. Complex predication: Who made the child pinch the elephant? In L. Uyechi and L. H. Wee, eds., *Reality Exploration and Discovery: Pattern Interaction in Language and Life*, pages 231–256. Stanford: CSLI Publications.

Butt, Miriam and Jafar Rizvi. 2010. Tense and Aspect in Urdu. In P. Cabredo-Hofherr and B. Laca, eds., *Layers of Aspect*. Stanford: CSLI Publications.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: The MIT Press.

Crouch, Dick, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. Maxwell III, and Paula Newman. 2011. *XLE Documentation*. Palo Alto Research Center.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, vol. 34 of *Syntax and Semantics*. Academic Press.

Hautli, Annette, Özlem Çetinoglu, and Josef van Genabith. 2010. Closing the Gap Between Stochastic and Hand-crafted LFG Grammars. In M. Butt and T. H. King, eds., *Proceedings of LFG10*, pages 270–289. CSLI Publications.

Hautli, Annette and Sebastian Sulger. 2011. Extracting and Classifying Urdu Multiword Expressions. In *Proceedings of the ACL 2011 Student Session*, pages 24–29. Portland, USA.

King, Tracy Holloway, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald Kaplan. 2003. The PARC700 Dependency Bank. In *Proceedings of the EACL03: 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*.

King, Tracy Holloway, Stefanie Dipper, Anette Frank, Jonas Kuhn, and John T. Maxwell III. 2000. Ambiguity Management in Grammar Writing. In *Proceedings of ESSLLI 2000*.

Kiparsky, Paul and Johan F. Staal. 1969. Syntactic and Semantic Relations in Paṇini. *Foundations of Language* 5(1):83–117.

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)*.

Palmer, Martha, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2007. Hindi Syntax: Annotating Dependency, Lexical Predicate-Argument Structure, and Phrase Structure. In *Proceedings of ICON'07: 7th International Conference on Natural Language Processing*, pages 259–268.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31(1):71–106.