A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu

Rajesh Bhatt
U. of Massachusetts
Amherst, MA, USA
bhatt@linguist.umass.edu

Bhuvana Narasimhan
U. of Colorado
Boulder, CO, USA
narasimb@colorado.edu

Martha Palmer
U. of Colorado
Boulder, CO, USA
mpalmer@colorado.edu

Owen Rambow
Columbia University
New York, NY, USA
rambow@ccls.columbia.edu

Dipti Misra Sharma
Int'l Institute of Info. Technology
Hyderabad, India
dipti@iiit.ac.in

Fei Xia
University of Washington
Seattle, WA, USA
fxia@u.washington.edu

Abstract

This paper describes the simultaneous development of dependency structure and phrase structure treebanks for Hindi and Urdu, as well as a PropBank. The dependency structure and the PropBank are manually annotated, and then the phrase structure treebank is produced automatically. To ensure successful conversion the development of the guidelines for all three representations are carefully coordinated.

1 Introduction

Annotated corpora have played an increasingly important role in the training of supervised natural language processing components. Today, treebanks have been constructed for many languages, including Arabic, Chinese, Czech, English, French, German, Korean, Spanish, and Turkish. This paper describes the creation of a Hindi/Urdu ***multi-representational and multi-layered treebank***. *Multi-layered* means that we design the annotation process from the outset to include both a syntactic annotation and a lexical semantic annotation such as the English PropBank (Palmer et al. 2005). *Multi-representational* means that we distinguish conceptually *what* is being represented from *how* it is represented; for example, in a case of long-distance *wh*-movement in English as in *Who do you think will come,* we can choose to represent the fact that *who* is an argument of *come*, or not (*what* to represent). Having made this choice, we can determine *how* to represent it: For example, we can use a discontinuous constituent (crossing arcs), or we can use a trace and co-indexation.

Flexibility of representation is important because the proper choice of representation of the syntax of a language is itself an issue in parsing research. In the application of the Collins parser to the Prague Dependency Treebank (Collins et al. 1999) the automatic mapping from dependency to phrase-structure was a major area of research. Similarly, automatically changing the representation in a phrase structure treebank can also improve parsing results (for example Klein & Manning 2003). Finally, there is increasing interest in the use of dependency parses in NLP applications, as they are considered to be simpler structures which can be computed more rapidly and are closer to the kinds of semantic representations that applications can make immediate use of (McDonald et al. 2005, CoNLL 2006 Shared Task). We first provide a comparison of dependency structure and phrase structure in Section 2. Section 3 describes our treebank, Section 4 explores language-specific linguistic issues that require special attention to ensure consistent conversion, and Section 5 summarizes our conversion approach.

2 Two Kinds of Syntactic Structure

Two different approaches to describing syntactic structure, dependency structure (DS) (Mel'čuk 1979) and phrase structure (PS) (Chomsky, 1981), have in a sense divided the field in two, with parallel efforts on both sides. Formally, in a PS tree, all and only the leaf nodes are labeled

Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP 2009, pages 186–189,
Suntec, Singapore, 6-7 August 2009.

with words from the sentence (or empty categories), while the interior nodes are labeled with nonterminal labels. In a dependency tree, all nodes are labeled with words from the sentence (or empty categories). Linguistically, a PS groups consecutive words hierarchically into phrases (or constituents), and each phrase is assigned a syntactic label. In a DS, syntactic dependency (i.e., the relation between a syntactic head and its arguments and adjuncts) is the primary syntactic relation represented. The notion of constituent is only derived.

In a dependency representation, a node stands for itself, for the lexical category (or "preterminal") spanning only the word itself (e.g., N), and for its maximal projection spanning the node and all words in the subtree it anchors (e.g., NP). Thus, intermediate projections which cover only some of the dependents of a word (such as N' or VP) do not directly correspond to anything in a dependency representation. Attachments at the different levels of projection are therefore not distinguished in a dependency tree. This has certain ramifications for annotation. Conisder for example scope in conjunctions. The two readings of *young men and women* can be distinguished (are the women young as well or not?). If a dependency representation represents conjunction by treating the conjunction as a dependent to the first conjunct, then the two readings do not receive different syntactic representations, unless a scope feature is introduced for the adjective. Suppose *y* depends on *x* in a DS, we need to address the following questions in order to devise a DS-to-PS conversion algorithm that builds the corresponding phrase structure: *1) What kinds of projections do x and y have? 2) How far should y project before it attaches to x's projection? 3) What position on x's projection chain should y's projection attach to?* These questions are answered by the annotation manual of the target PS representation – there are many possible answers. If the source dependency representation contains the right kind of information (for example, the scope of adjectives in conjunctions), and if the target phrase structure representation is well documented, then we can devise a conversion algorithm.

Another important issue is that of "non-projectivity" which is used to represent discontinuous constituents. Non-projectivity is common in dependency-based syntactic theories, but rare in phrase structure-based theories. The next section highlights our most salient representation choices in Treebank design.

3 Treebank Design

Our goal is the delivery of a treebank that is *multi-representational*: it will have a syntactic dependency version and a phrase structure version. Another recent trend in treebanking is the addition of deeper, semantic levels of annotation on top of the syntactic annotations of the PTB, for example PropBank (Palmer et al. 2005). A multi-layered approach is also found in the Prague Dependency Treebank (Hajič et al. 2001), or in treebanks based on LFG (King et al. 2003) or HPSG (Oepen et al. 2002). A lesson learned here is that the addition of deeper, more semantic levels may be complicated if the syntactic annotation was not designed with the possibility of multiple layers of annotation in mind. We therefore also propose a treebank that is from the start *multi-layered*: we will include a PropBank-style predicate-argument annotation in the release. Crucially, the lexical subcategorization frames that are made explicit during the process of propbanking should always inform the syntactic structure of the treebanking effort. In addition, some of the distinctions made by PS that are not naturally present in DS, such as unaccusativity and null arguments, are more naturally made during PropBank annotation. Our current approach anticipates that the addition of the PropBank annotation to the DS will provide a rich enough structure for accurate PS conversion.

In order to ensure successful conversion from DS to PS, we are simultaneously developing three sets of guidelines for Hindi: dependency structure, phrase structure, and PropBank. While allowing DS and PS guidelines to be based on different, independently motivated principles (see Section 4), we have been going through a comprehensive list of constructions in Hindi, carefully exploring any potentially problematic issues. Specifically, we make sure that both DS and PS represent the same syntactic facts *(what is represented)*: we know that if PS makes a distinction that neither DS nor PropBank make, then we cannot possibly convert automatically. Furthermore, we coordinate the guidelines for DS and PS with respect to the examples chosen to support the conversion process. These examples form a conversion test suite.

4 Syntactic Annotation Choices

4.1 Dependency Structure Guidelines

Our dependency analysis is based on the Paninian grammatical model (Bharati et al 1999, Sharma et al. 2007). The model offers a syntactico-semantic level of linguistic knowledge with an especially transparent relationship between the syntax and the semantics. The sentence is treated as a series of modifier-modified relations which has a primary modified (generally the main verb). The appropriate syntactic cues (relation markers) help in identifying various relations. The relations are of two types – karaka and others. 'Karakas' are the roles of various participants in an action (arguments). For a noun to hold a karaka relation with a verb, it is important that they (noun and verb) have a direct syntactic relation. Relations other than 'karaka' such as purpose, reason, and possession are also captured using the relational concepts of the model (adjuncts). These argument labels are very similar in spirit to the verb specific semantic role labels used by PropBank, which have already been successfully mapped to richer semantic role labels from VerbNet and FrameNet. This suggests that much of the task of PropBanking can be done as part of the dependency annotation.

4.2 Phrase Structure Guidelines

Our PS guidelines are inspired by the Principles-and-Parameters methodology, as instantiated by the theoretical developments starting with Government and Binding Theory (Chomsky 1981). We assume binary branching. There are three theoretical commitments/design considerations that underlie the guidelines. First, any minimal clause distinguishes at most two positions structurally (the core arguments). These positions can be identified as the specifier of VP and the complement of V. With a transitive predicate, these positions are occupied by distinct NPs while with an unaccusative or passive, the same NP occupies both positions. All other NPs are represented as adjuncts. Second, we represent any displacement of core arguments from their canonical positions, irrespective of whether a clause boundary is crossed, via traces. The displacement of other arguments is only represented if a clause boundary is crossed. Third, syntactic relationships such as agreement and case always require c-command but do not necessarily require a [specifier, head] configuration. Within these constraints, we always choose the simplest structure compatible with the word order. We work with a very limited set of category labels (NP, AP, AdvP, VP, CP) assuming that finer distinctions between different kinds of verbal functional heads can be made via features.

4.3 Two Constructions in Hindi

We give examples for two constructions in Hindi and show the DS and PS for each.

Simple Transitive Clauses:

(1) raam-ne khiir khaayii
ram-erg rice-pudding ate
'Ram ate rice-pudding.'

The two main arguments of the Hindi verb in Figure 1(b) have dependency types k1 and k2. They correspond roughly to subject and object, and they are the only arguments that can agree with the verb. In the PS, Figure 1(a), the two arguments that correspond to k1 and k2 have fixed positions in the phrase structure as explained in Section 4.2.

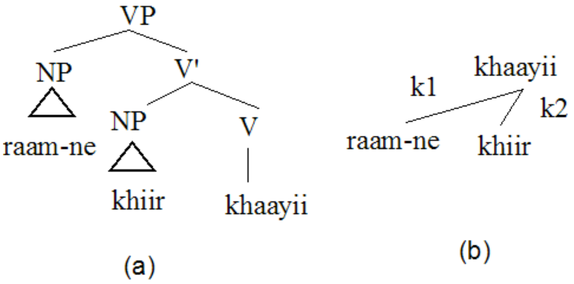

Figure 1: PS and DS for transitive clause in (1).

Unaccusative verbs:

(2) darwaazaa khul rahaa hai
door.M open Prog.MSg be.Prs.Sg
'The door is opening.'

Here, the issue is that the DS guidelines treats unaccusatives like other intransitives, with the surface argument simply annotated as k1. In contrast, PS shows a derivation in which the subject originates in object position.

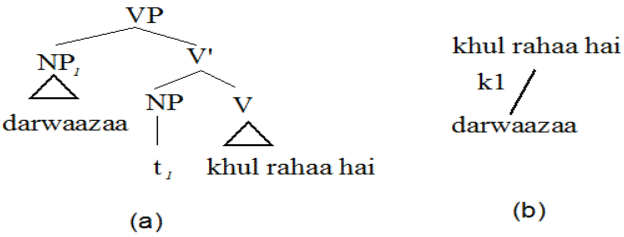

Figure 2: PS and DS for the unaccusative in (2).

5 Conversion Process

The DS-to-PS conversion process has three steps. First, for each (DS, PS) pair appearing in the conversion test suite, we run a consistency checking algorithm to determine whether the DS and the PS are consistent. The inconsistent cases are studied manually and if the inconsistency cannot be resolved by changing the analyses used in the guidelines, a new DS that is consistent with the PS is proposed. We call this new dependency structure "DS_{cons}" ("cons" for "consistency"; DS_{cons} is the same as DS for the consistent cases). Because the DS and PS guidelines are carefully coordinated, we expect the inconsistent cases to be rare and well-motivated. Second, conversion rules are extracted automatically from these (DS_{cons}, PS) pairs. Last, given a new DS, a PS is created by applying conversion rules. Note that non-projective DSs will be converted to projective DS_{cons}. (For an alternate account of handling non-projective DSs, see Kuhlman and Möhl (2007).) A preliminary study on the English Penn Treebank showed promising results and error analyses indicated that most conversion errors were caused by ambiguous DS patterns in the conversion rules. This implies that including sufficient information in the input DS could reduce ambiguity, significantly improving the performance of the conversion algorithm. The details of the conversion algorithm and the experimental results are described in (Xia et al., 2009).

6 Conclusion

We presented our approach to the joint development of DS and PS treebanks and a PropBank for Hindi/Urdu. Since from the inception of the project we have planned manual annotation of DS and automatic conversion to PS, we are developing the annotation guidelines for all structures in parallel. A series of linguistic constructions with specific examples are being carefully examined for any DS annotation decisions that might result in inconsistency between DS and PS and/or multiple conversion rules with identical DS patterns. Our preliminary studies yield promising results, indicating that coordinating the design of DS/PS and PropBank guidelines and running the conversion algorithm in the early stages is essential to the success of building a multi-representational and multi-layered treebank.

Acknowledgments

This work is supported by NSF grants CNS-0751089, CNS-0751171, CNS-0751202, and CNS-0751213.

References

A. Bharati, V. Chaitanya and R. Sangal. 1999. *Natural Language Processesing: A Paninian Perspective*, Prentice Hall of India, New Delhi.

N. Chomsky. 1981. *Lectures on Government and Binding: The Pisa Lectures*. Holland: Foris Publications.

M. Collins, Jan Hajič, L. Ramshaw and C. Tillmann. 1999. A Statistical Parser for Czech. *In the Proc of ACL-1999,* pages 505-512.

J. Hajič, E. Hajicova, M. Holub, P. Pajas, P. Sgall, B. Vidova-Hladka, and V. Reznickova. 2001. The Current Status of the Prague Dependency Treebank. *Lecture Notes in Artificial Intelligence (LNAI)* 2166, pp 11—20, NY.

T. H. King, R. Crouch, S. Riezler, M. Dalrymple and R. Kaplan. 2003. The PARC700 Dependency Bank. *In Proc. of the 4th Int' Workshop on Linguistically Interpreted Corpora (LINC-2003),* Budapest, Hungary.

D. Klein and C. D. Manning. 2003. Accurate Unlexicalized Parsing. *In the Proc of ACL-2003,*.Japan

M. Kuhlmann and M. Möhl. 2007. Mildly context-sensitive dependency language. *In the Proc of ACL 2007.* Prague, Czech Republic.

R. McDonald, F. Pereira, K. Ribarov and J. Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. *In Proc. of HLT-EMNLP 2005.*

I. Melčuk. 1979. *Studies in Dependency Syntax*. Karoma Publishers, Inc.

S. Oepen, K. Toutanova, S. M. Shieber, C. D. Manning, D. Flickinger, and T. Brants, 2002. The LinGO Redwoods Treebank: Motivation and Preliminary Applications. *In Proc. of COLING, 2002.* Taipei, Taiwan.

M. Palmer, D. Gildea, P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles.*Computational Linguistics, 31(1):*71-106.

D. M. Sharma, R. Sangal, L. Bai, R. Begam, and K.V. Ramakrishnamacharyulu. 2007. *AnnCorra : TreeBanks for Indian Languages,* Annotation Guidelines (manuscript), IIIT, Hyderabad, India.

F. Xia, O. Rambow, R. Bhatt, M. Palmer and D. Sharma, 2009. Towards a Multi-Representational Treebank. *In Proc. of the 7th Int'lWorkshop on Treebanks and Linguistic Theories (TLT-7).*