A Dependency Treebank of Urdu and its Evaluation

Riyaz Ahmad Bhat
LTRC, IIIT Hyderabad
riyaz.bhat@research.iiit.ac.in

Dipti Misra Sharma
LTRC, IIIT Hyderabad
dipti@iiit.ac.in

Abstract

In this paper we describe a currently underway treebanking effort for Urdu-a South Asian language. The treebank is built from a newspaper corpus and uses a Karaka based grammatical framework inspired by Paninian grammatical theory. Thus far 3366 sentences (0.1M words) have been annotated with the linguistic information at morpho-syntactic (morphological, part-of-speech and chunk information) and syntactico-semantic (dependency) levels. This work also aims to evaluate the correctness or reliability of this manual annotated dependency treebank. Evaluation is done by measuring the inter-annotator agreement on a manually annotated data set of 196 sentences (5600 words) annotated by two annotators. We present the qualitative analysis of the agreement statistics and identify the possible reasons for the disagreement between the annotators. We also show the syntactic annotation of some constructions specific to Urdu like $Ezafe$ and discuss the problem of word segmentation (tokenization).

1 Introduction

Hindi and Urdu[1] are often socially considered distinct language varieties, but linguistically the division between the two varieties is not well-founded. (Masica, 1993, p. 27) explains that while they are different languages officially, they are not even different dialects or sub-dialects in a linguistic sense; rather, they are different literary styles based on the same linguistically defined sub-dialect. He further explains that at colloquial level, Hindi and Urdu are nearly identical, both in terms of core vocabulary and grammar. However, at formal and literary levels, vocabulary differences begin to loom much larger (Hindi drawing its higher lexicon from Sanskrit and Urdu from Persian and Arabic) to the point where the two styles/languages become mutually unintelligible. In written form not only lexical items but the way Urdu and Hindi is written makes one believe that they are two separate languages. They are written in separate orthographies, Hindi being written in Devanagari, and Urdu in a modified Perso-Arabic script. Under the treebanking effort for Indian languages, two separate treebanks are being built for both Hindi and Urdu. Among the two, however, Hindi treebank has matured and grown considerably (Bhatt et al., 2009), (Palmer et al., 2009).

The paper is arranged as follows, next Section gives a brief overview of the related works on syntactic treebanking. Section 3 describes the grammatical formalism chosen for the annotation. In Section 4 we discuss treebanking pipeline of Urdu followed by some of the Urdu specific issues in Section 5. In Section 6 we discuss the empirical results of inter-annotator agreement. Section 7, concludes the paper.

2 Related Work

A treebank is a text corpus annotated with syntactic, semantic and sometimes even inter sentential relations (Hajičová et al., 2010). Treebanks are of multifold importance, they are an invaluable resource for testing linguistic theories on which they are built

[1] Hindi-Urdu is an Indo-Aryan language spoken mainly in North India and Pakistan.

Proceedings of the 6th Linguistic Annotation Workshop, pages 157–165,
Jeju, Republic of Korea, 12-13 July 2012. ©2012 Association for Computational Linguistics

and are used for a number of NLP tasks like training and testing syntactic parsers. Owing to their great importance, a number of syntactic treebanking projects have been initiated for many different languages. Among the treebanks include Penn treebank (PTB) (Marcus et al., 1993), Prague Dependency treebank (PDT) (Hajicová, 1998) for Czech, (Rambow et al., 2002) for English, Alpino (Van der Beek et al., 2002) for Dutch, TUT (Bosco and Lombardo, 2004) for Italian, TIGER (Brants et al., 2002) for German and many others. Currently existing treebanks mainly differ in the grammatical formalism adopted. Dependency based formalism compared with the constituency based formalism is assumed to suit better for representing syntactic structures of free word order languages, its representation does not crucially rely on the position of a syntactic unit in a sentence thus easily handles the scrambling of arguments in such languages (Shieber, 1985), (Bharati et al., 1995), (Hajič, 1998), (Hajicová, 1998), (Oflazer et al., 2003). Not only are dependency-based representations suitable for less configurational languages, they are also favorable for a number of natural language processing applications (Culotta and Sorensen, 2004), (Reichartz et al., 2009).

Structural relations like subject and direct object are believed to be less relevant for the grammatical description of Indian languages (ILs) because of the less configurational nature of these languages (Bhat, 1991). Indian languages are morphologically rich and have a relatively free constituent order. (Begum et al., 2008) have argued in favor of using Karaka relations instead of structural relations for the syntactic analysis of ILs. They proposed an annotation scheme for the syntactic treebanking of ILs based on the Computational Paninian Grammar (CPG), a formalism inspired by Paninian grammatical theory. Currently dependency treebanks of four ILs, namely Hindi, Urdu, Bangla and Telegu, are under development following this annotation scheme. The dependency structures in all the four treebanks are, under this annotation scheme, annotated with the Karaka relations. Although English does not belong to the free word order languages, a number of attempts have been made to study the applicability of CPG based syntactic analysis to it as well (Bharati et al., 1996), (Vaidya et al., 2009), (Chaudhry and Sharma, 2011).

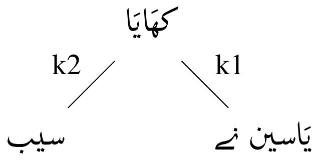
3 CPG Formalism

The CPG formalism, inspired by the grammatical theory of Panini, the fifth century B.C. grammarian of Sanskrit, is a dependency grammar. As in other dependency grammars, the syntactic structures in this formalism essentially consists of a set of binary, asymmetric relations between words of a sentence. A dependency relation is defined between a dependent, a syntactically subordinate word and a head word on which it depends. In this formalism verb is treated as the primary modified (the root of the dependency tree) and the elements (nominals) modifying the verb participate in the activity specified by it. The relation that holds between a verb and its modifier is called a *karaka* relation. There are six basic *karakas* defined by Panini namely (i) *karta* 'agent', (ii) *karma* 'theme', (iii) *karana* 'instrument', (iv) *sampradaan* 'recipient', (v) *apaadaan* 'source', and (vi) *adhikarana* 'location'. Besides *karaka* relations that hold between a verb and the participants of the action specified by the verb, dependency relations also exist between nouns (genitives), between nouns and their modifiers (adjectival modification, relativization), between verbs and their modifiers (adverbial modification including clausal subordination). A detailed tag-set containing all these different kinds of dependency relations has been defined in the annotation scheme based on the CPG formalism (Bharati et al., 2009). Examples (1) and (2) depict some of the *karaka* relations (*k1 'karta', k2 'karma', k3 'karana'*) of verbs کھَایَا *'eat'* and کَاٹَا *'cut'* respectively while example (3) shows a genitive relation between two nouns, یَاسین *'Yasin'* and قلم *'pen'*.

(1) یَاسین نے سیب کھَایَا

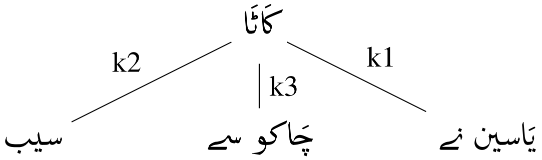
yAsIn-ne	*saeb*	*khAyA*
Yasin-ERG	apple-NOM	eat-PST+PERF

'Yasin ate an apple.'



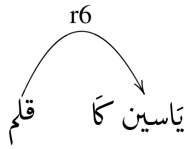
(2) یَاسِین نے چَاکو سے سَیب کَاٹَا

yAsIn-ne *chAku-se* *saeb*
Yasin-ERG knife-INST apple-NOM
kAtA
eat-PST+PERF
'Yasin cut the apple with a knife.'



(3) یَاسِین کَا قلم

yAsIn-kA *qalam*
Yasin-GEN pen
'Yasin's pen.'



4 Annotation Pipeline

The dependency treebanks for Indian languages based on CPG formalism are developed following a generic pipeline. The process of treebank development under the pipeline consists of a series of steps namely (i) Tokenization, (ii) Morph-Analysis, (iii) POS-tagging, (iv) Chunking, and (v) Dependency annotation. Annotation process begins with the tokenization of raw text. The tokens obtained during tokenization are, in the next steps, annotated with morphological and POS tag information. After morph-analysis and POS-tagging correlated, inseparable words are grouped into chunks. The processing at the steps mentioned thus far are automated by highly accurate tools built in-house (tokenizer, morph analyzer, POS-tagger and chunker). The output of each tool is, however, manually corrected and validated by the human annotators. The final step in the pipeline is the manual dependency annotation. Only the inter-chunk dependencies are marked leaving the dependencies between words in a chunk unspecified because the intra-chunk dependencies are observed to be highly predictive given the head of a chunk and can be easily generated by a set of rules at a later stage.

UDT is steadily being developed following this treebanking pipeline by annotating the newspaper articles by a team of annotators with expertise in linguistics. The tool being used for the annotation is a part of Sanchay[2] (Singh, 2006). The annotations are represented in Shakti Standard Format (SSF) (Bharati et al., 2007). Hitherto, 3226 sentences (around 0.1M words) have been annotated with dependency structure. Each sentence contains an average of 29 words and an average of 13.7 chunks of average length 2.0.

5 Languages Specific Issues

5.1 Word segmentation

Urdu is written in a *Nastaliq style cursive Arabic script*. In this script an individual letter acquires different shapes upon joining with the adjacent letters. There are four possible shapes a letter can acquire namely $initial, medial, final$ form in a connected sequence of letters or an $isolated$ form. The letters acquiring all these four shapes depending on the context of their occurrence are called as $joiners$. An another set of letters, however, called as $non - joiners$ do not adhere to this four-way shaping. They only join with the letters before them and have only $final$ and $isolated$ forms. An example of a joiner is Arabic Letter 'Teh' ت and a non-joiner is Arabic letter '$waaw$' و.

The concept of space as a word boundary marker is not present in Urdu writing (Durrani and Hussain, 2010), (Lehal, 2010). Space character is primarily required to generate correct shaping of words. For example a space is necessary within the word ضَرُورَت مَند "needy" to generate the visually correct and acceptable form of this word. Without

[2] http://apps.sanchay.co.in/latest-builds/

space it appears as ضَرُورَتمَند which is visually incorrect. In contrast to this, writers of Urdu find it unnecessary to insert a space between the two words اردُو مَرکز "Urdu Center", because the correct shaping is produced automatically as the first word ends with a non-joiner. Therefore اردُو مَرکز and اردُومَرکز look identical. Although space character is primarily used to generate correct shapes of words, it is now being used as a word separator as well. This two-way function of space character in Urdu makes it an unreliable cue for word boundary which poses challenges to the process of tokenization. In UDT pipeline raw text is tokenized into individual tokens using a tokenizer which uses space as word boundary. The generation of erroneous tokens (single words broken into multiple fragments) is obvious, since, as mentioned above, space not only marks word boundary it is also used to generate correct shaping of a word. To ensure that only valid tokens are processed in the further stages of the pipeline, tokenization is followed by human post-editing. The fragments of a word are joined using an underscore '_'. This ensures that such words retain their visually correct shape. For example two fragments ضَرُورَت and مَند of a single word ضَرُورَت مَند generated by the tokenizer will be joined into single word with an '_' as ضَرُورَت_مَند.

5.2 Ezafe

Ezafe is an enclitic short vowel e which joins two nouns, a noun and an adjective or an adposition and a noun into a possessive relationship. In Urdu ezafe is a loan construction from Persian, it originated from an Old Iranian relative pronoun $-hya$, which in Middle Iranian changed into y/i a device for nominal attribution (Bögel et al., 2008). The Urdu ezafe construction functions similarly to that of its Persian counter part. In both the languages the ezafe construction is head-initial which is different from the typical head-final nature of these languages. As in Persian the Urdu ezafe lacks prosodic independence, it is attached to a word to its left which is the head of the ezafe construction. It is pronounced as a unit with the head and licenses a modifier to its right. This is in contrast to the Urdu genitive construction, which conforms to the head-final pattern typical for Urdu. The genitive marker leans on the modifier of the genitive construction not on the head and is pronounced as a unit with it. Example (4) is a typical genitive construction in Urdu while (5) shows an ezafe construction.

(4) یَاسِین کَا قلم

yAsIn-kA	*qalam*
Yasin-GEN	pen

'Yasin's pen.'

(5) حکومتِ پَاکستَان

hukummat-e	*Pakistan*
government-Ez	Pakistan

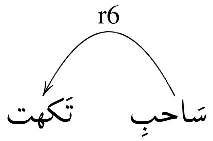
'Government of Pakistan.'

The ezafe construction in Urdu can also indicate relationships other than possession. In current Urdu treebank when an ezafe construction is used to show possessive relationship, it is annotated similar to genitive constructions indicating possession with an *"r6"* label as shown in example (6), the head noun سَاحب *'owner' 'possesses'* the modifying noun تَکھت *'throne'*. However, in example (7) ezafe does not indicate a possessive meaning, in such cases *"NMOD"* (noun modifier) is used instead of *"r6"*, the adjective روشن *'bright'* does not stand in a possession relation to the روزِ *'day'*, but simply modifies the head noun in an attributive manner.

(6) سَاحبِ تَکھت

sahb-e	*takht*
owner-Ez	throne

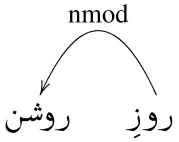
'The owner of the throne.'



(7) روزِ روشن

rooz-e	*rooshan*
day-Ez	bright

'Bright day.'



6 Agreement Analysis

In order to ensure the reliability of manual dependency annotations in UDT, we did an agreement analysis using a data set of 5600 words annotated by two annotators, without either annotator knowing other's decisions. A good agreement on the data set will assure that the annotations in UDT are reliable. The data set used contains 2595 head-dependent dependency chains marked with dependency relations belonging to a tag-set of 39 tags. The agreement measured is chunk based; for each chunk in a sentence agreement was measured with regard to its relation with the head it modifies.

Inter-annotator agreement was measured using Cohen's kappa (Cohen and others, 1960) which is the mostly used agreement coefficient for annotation tasks with categorical data. Kappa was introduced to the field of computational linguistics by (Carletta et al., 1997) and since then many linguistics resources have been evaluated using the matrix such as (Uria et al., 2009), (Bond et al., 2008), (Yong and Foo, 1999). The kappa statistics show the agreement between the annotators and the reproducibility of their annotated data sets. Similar results produced by the annotators on a given data set proves the similarity in their understanding of the annotation guidelines. However, a good agreement does not necessarily ensure validity, since annotators can make similar kind of mistakes and errors.

The kappa coefficient κ is calculated as:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{8}$$

$Pr(a)$ is the observed agreement among the coders, and $Pr(e)$ is the expected agreement, that is, $Pr(e)$ represents the probability that the coders agree by chance.

Based on the interpretation matrix of kappa value proposed by Landis and Koch (Landis and Koch, 1977) as presented in Table 1, we consider that the agreement as presented in Table 2, between the annotators on the data set used for the evaluation, is reliable. There is a substantial amount of agreement between the annotators which implies their similar understanding of the annotation guidelines and of the linguistic phenomenon present in the language.

Kappa Statistic	Strength of agreement
<0.00	Poor
0.0-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1.00	Almost perfect

Table 1: Coefficients for the agreement-rate based on (Landis and Koch, 1977).

No. of Annotations	Agreement	Pr(a)	Pr(e)	Kappa
2595	1921	0.74	0.097	0.71

Table 2: Kappa statistics

Urdu as discussed earlier is a morphologically rich language, information concerning the arrangement of words into syntactic units or cues to syntactic relations, is expressed at word level through case clitics (Mohanan, 1990). Because information about the relations between syntactic elements is expressed at word level, the prediction of the syntactic relations becomes easier for an annotator. However, as mentioned in Table 3 case markers and case roles don not have a one to one mapping, each case marker is distributed over a number of case roles, this phenomenon is called as **case syncretism**. Among the 6 case markers viz نے $(ergative)$, کو $(dative)$, کو $(accusative)$, سے $(instrumental)$, سے $(ablative)$, کَا $(genitive)$ and مے, پر $(locative)$ only نے $(ergative)$ is unambiguous, all others are ambiguous between different roles. This syncretism is one of the reason for the disagreement between the annotators. Out of 965 case marked nominals 735 are agreed upon by both the annotators and for 230 nominals both disagreed. Examples below show syncretism in case marker کو 'ko'. کو marks the 'recipient', 'theme' and the 'experiencer' of the main verbs in sentences (9), (10) and (11) respectively.

	نے (ne)	کو (ko)	کَا (kA)	سے (se)	مے (mem)	پر (par)
$k1$	100	22	1	0	0	0
$k2$	0	46	1	15	0	0
$k3$	0	0	0	2	0	0
$k4$	0	17	0	19	0	0
$k4a$	0	2	0	0	0	0
$k5$	0	0	0	14	0	0
$k7$	0	0	1	1	60	70
$k7t$	0	5	2	11	6	0
$k7p$	0	0	0	0	19	10
$r6$	0	0	89	0	0	0
rh	0	0	0	5	0	0

Table 3: Agreement among the Annotators on Karaka roles given a Case Marker.

The nominals carrying کو in these sentences will be labeled in UDT as *k4 'recipient'*, *k2 'theme'* and *k4a 'experiencer'* respectively.

(9) نَادیَا نے یَاسین کو کتَاب دی

Nadiya-ne *Yasin-ko* *kitab*
Nadya-ERG Yasin-DAT book-NOM
di.
give-PST+PRF
'Nadiya gave Yasin a book.'

(10) نَادیَا نے یَاسین کو بُلَایَا

Nadiya-ne *Yasin-ko* *bhulaayaa.*
Nadya-ERG Yasin-ACC call-PST+PRF
'Nadiya called Yasin.'

(11) یَاسین کو کہَانی یَاد آی

Yasin-ko *kahani* *yaad*
Yasin-Dat story-NOM memory
aayi.
come-PST+PRF
'Yasin remembered the story.'

Table 5 shows the statistics of the annotation-the number of labels used by each annotator and the frequency of agreement and disagreement per label. Statistics in Table 4 and 5 show that a considerable amount of confusion is between *'k1' (agent) and 'k2' (theme)*; *'k1' (agent) and 'pof' (part of)*; *'k1s' (noun complement) and 'pof' (part of)* and *'k2' (theme) and 'pof' (part of)*. Out of 110 disagreements for label *'pof'*, the annotators differ 81 (74%) times in marking a given dependency structure either with a *'pof'* relation or with *'k1, 'k1s' or 'k2'*. Similarly for *'k1'* 38% disagreements are between *'k2' and 'pof'* and for *'k2'* 49% disagreements are between *'k1' and 'pof'*. The high number of disagreements among the members of this small subset of labels $(k1, k2, k1s, pof)$ suggest the validity of the disagreement that is to say that the disagreements are not random or by chance and can be attributed to the ambiguity or some complex phenomenon in the language. All the disagreements involving *'pof'* relation occur due to the complexity of identifying the complex predicates in Urdu. The challenges in the identification of complex predicates (Begum et al., 2011) coupled with similar syntactic distribution of these Karaka roles explain the differences among the annotators for these relations. Take for example the case of sentences (12) and (13) both مدَد *'help'* and چَابی *'key'* have similar syntactic context, but in (12) مدَد *'help'* is part of the complex predicate and has a *'pof'* (part of complex predicate) relation with the light verb لی *'take'* while in (13) چَابی *'key'* is the 'theme' of the main verb لی *'take'* and will be marked as its *'k2'*. Similarly in (14) and (15) دھمکی *'threat'* and کتَاب *'book'* have similar context, similar to مدَد *'help'* in (12), دھمکی *'threat'* has a *'pof'* relation with the verb دی *'give'* and کتَاب *'book'* in (15) is its 'theme' marked with the label *'k2'*.

(12) نَادیَا نے یَاسین سے مدَد لی

Nadiya-ne *Yasin-se* *madad*
Nadya-ERG Yasin-ABL help
li.
take-PST+PRF
'Nadiya took help from yasin.'

(13) نَادیَا نے یَاسین سے چَابی لی

Nadiya-ne *Yasin-se* *chaabi*
Nadya-ERG Yasin-ABL key-NOM
li.
take-PST+PRF
‘Nadiya took key from Yasin.’

(14) نَادیَا نے یَاسین کو دھمکی دی

Nadiya-ne *Yasin-ko* *dhamki*
Nadya-ERG Yasin-ACC threaten
di.
give-PST+PRF
‘Nadiya threatened Yasin.’

(15) نَادیَا نے یَاسین کو کتَاب دی

Nadiya-ne *Yasin-ko* *kitab*
Nadya-ERG Yasin-DAT book-NOM
di.
give-PST+PRF
‘Nadiya gave Yasin a book.’

	k1	**k1s**	**k2**	**k2s**	**k3**	**k4**	**k4a**	**k5**	**k7**	**k7p**	**k7t**	**pof**
k1	0	1	5	0	1	5	1	0	2	1	0	**11**
k1s	2	0	2	0	0	0	0	0	0	0	0	**16**
k2	**43**	2	0	1	0	1	0	3	2	0	1	**38**
k2s	0	1	8	0	0	0	0	0	0	0	0	2
k3	0	0	1	0	0	0	0	0	0	0	0	0
k4	2	0	6	0	0	0	1	0	0	0	0	0
k4a	1	0	0	0	0	0	0	0	0	0	0	0
k5	0	0	1	0	1	2	0	0	0	3	0	0
k7	0	0	0	0	0	0	0	0	0	3	3	1
k7p	0	0	0	0	0	0	0	0	8	0	0	0
k7t	0	0	0	0	0	0	0	0	2	0	0	0
pof	1	**9**	6	1	0	0	0	0	2	0	0	0

Table 4: Confusion Matrix between the Annotators.

7 Conclusion

In this paper we have discussed an ongoing effort of building a dependency treebank for Urdu based on CPG framework. We discussed some of the Urdu specific issues like $Ezafe$ construction and word segmentation encountered during the treebank development. We also discussed the evaluation of dependency level annotation by measuring the inter-annotator agreement using the Kappa statistics. The

	Relations	**Ann.1**	**Ann.2**	**Agr.**	**Disagr.**
1	$ras-k4$	0	1	0	1
2	$ras-k1$	4	6	3	4
3	$ras-k2$	1	3	0	4
4	$pof\_\_idiom$	1	0	0	1
5	$r6-k1$	10	8	4	10
6	$r6-k2$	63	50	43	27
7	$rbmod$	2	0	0	2
8	pof	325	271	243	110
9	rt	43	48	38	15
10	$k3$	11	8	6	7
11	rs	1	8	1	7
12	$k2s$	21	30	17	17
13	$k2p$	4	3	2	3
14	$k1$	346	320	254	158
15	rd	13	3	2	12
16	$k2$	249	298	179	189
17	$nmod\_\_relc$	27	30	13	31
18	$k7$	160	156	123	70
19	$jjmod$	23	8	8	15
20	$k5$	15	28	12	19
21	$k4$	46	50	34	28
22	$nmod\_\_k2inv$	2	3	2	1
23	rh	21	15	7	22
24	$k4a$	10	12	7	8
25	$k7a$	5	6	4	3
26	adv	47	45	30	32
27	$nmod\_\_k1inv$	0	1	0	1
28	$fragof$	6	7	5	3
29	$k7p$	46	44	29	32
30	$k7t$	67	71	53	32
31	$nmod\_\_emph$	1	2	0	3
32	$k1s$	62	70	41	50
33	$r6$	297	335	258	116
34	$k1u$	0	1	0	1
35	$vmod$	102	98	63	74
36	$nmod$	91	96	48	91
37	$ccof$	436	486	389	144
38	$sent-adv$	1	0	0	1
39	$r6v$	5	5	3	4

Table 5: Agreement and Disagreement between the Annotators.

agreement as presented in this work is considered to be reliable and substantial ensuring that the syntactic annotations in the treebank are consistent and are annotated by the annotators with a substantial clarity of the annotation guidelines.

8 Acknowledgement

We would like to thank Anjum Parveen for annotating the data set used for the experiments of inter-annotator agreement. The work reported in this paper is supported by the NSF grant (Award Number: CNS 0751202; CFDA Number: 47.070). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

R. Begum, S. Husain, A. Dhwaj, D.M. Sharma, L. Bai, and R. Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of IJCNLP*. Citeseer.

R. Begum, K. Jindal, A. Jain, S. Husain, and D. Misra Sharma. 2011. Identification of conjunct verbs in hindi and its effect on parsing accuracy. *Computational Linguistics and Intelligent Text Processing*, pages 29–40.

A. Bharati, V. Chaitanya, R. Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural Language Processing: A Paninian Perspective*. Prentice-Hall of India.

A. Bharati, M. Bhatia, V. Chaitanya, and R. Sangal. 1996. Paninian grammar framework applied to english. Technical report, Technical Report TRCS-96-238, CSE, IIT Kanpur.

A. Bharati, R. Sangal, and D.M. Sharma. 2007. Ssf: Shakti standard format guide. Technical report, Technical report, IIIT Hyderabad.

A. Bharati, D.M. Sharma, S. Husain, L. Bai, R. Begum, and R. Sangal. 2009. Anncorra: Treebanks for indian languages guidelines for annotating hindi treebank (version–2.0).

D.N.S. Bhat. 1991. *Grammatical relations: the evidence against their necessity and universality*. Psychology Press.

R. Bhatt, B. Narasimhan, M. Palmer, O. Rambow, D.M. Sharma, and F. Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189. Association for Computational Linguistics.

T. Bögel, M. Butt, and S. Sulger. 2008. Urdu ezafe and the morphology-syntax interface. *Proceedings of LFG08*.

F. Bond, S. Fujita, and T. Tanaka. 2008. The hinoki syntactic and semantic treebank of japanese. *Language Resources and Evaluation*, 42(2):243–251.

C. Bosco and V. Lombardo. 2004. Dependency and relational structure in treebank annotation. In *Proceedings of Workshop on Recent Advances in Dependency Grammar at COLING'04*.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.

J. Carletta, S. Isard, G. Doherty-Sneddon, A. Isard, J.C. Kowtko, and A.H. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational linguistics*, 23(1):13–31.

H. Chaudhry and D.M. Sharma. 2011. Annotation and issues in building an english dependency treebank.

J. Cohen et al. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics.

N. Durrani and S. Hussain. 2010. Urdu word segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–536. Association for Computational Linguistics.

E. Hajicová. 1998. Prague dependency treebank: From analytic to tectogrammatical annotation. *Proceedings of TSD98*, pages 45–50.

J. Hajič. 1998. Building a syntactically annotated corpus: The prague dependency treebank. *Issues of valency and meaning*, pages 106–132.

E. Hajičová, A. Abeillé, J. Hajič, J. Mírovský, and Z. Urešová. 2010. Treebank annotation. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL. ISBN 978-1420085921.

J.R. Landis and G.G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

G.S. Lehal. 2010. A word segmentation system for handling space omission problem in urdu script. In *23rd International Conference on Computational Linguistics*, page 43.

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

C.P. Masica. 1993. *The Indo-Aryan Languages*. Cambridge Univ Pr, May.

T.W. Mohanan. 1990. *Arguments in Hindi*. Ph.D. thesis, Stanford University.

K. Oflazer, B. Say, D.Z. Hakkani-Tür, and G. Tür. 2003. Building a turkish treebank. *Abeillé (Abeillé, 2003)*, pages 261–277.

M. Palmer, R. Bhatt, B. Narasimhan, O. Rambow, D.M. Sharma, and F. Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

O. Rambow, C. Creswell, R. Szekely, H. Taber, and M. Walker. 2002. A dependency treebank for english. In *Proceedings of LREC*, volume 2.

F. Reichartz, H. Korte, and G. Paass. 2009. Dependency tree kernels for relation extraction from natural language text. *Machine Learning and Knowledge Discovery in Databases*, pages 270–285.

S.M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343.

L. Uria, A. Estarrona, I. Aldezabal, M. Aranzabe, A. Díaz de Ilarraza, and M. Iruskieta. 2009. Evaluation of the syntactic annotation in epec, the reference corpus for the processing of basque. *Computational Linguistics and Intelligent Text Processing*, pages 72–85.

A. Vaidya, S. Husain, P. Mannem, and D. Sharma. 2009. A karaka based annotation scheme for english. *Computational Linguistics and Intelligent Text Processing*, pages 41–52.

L. Van der Beek, G. Bouma, R. Malouf, and G. Van Noord. 2002. The alpino dependency treebank. *Language and Computers*, 45(1):8–22.

C. Yong and S.K. Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation.