A Karaka Based Annotation Scheme for English

by

Ashwini Vaidya, Samar Husain, Prashanth Reddy, Dipti M Sharma

in

In Proceedings of the CICLing-2009, Mexico City, Mexico. 2009.

Report No: IIIT/TR/2009/2

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2009

A Karaka Based Annotation Scheme for English

Ashwini Vaidya, Samar Husain, Prashanth Mannem, and Dipti Misra Sharma

Language Technologies Research Centre, International Institute of Information Technology, Hyderabad, India
{ashwini_vaidya, samar, prashanth}@research.iiit.ac.in, dipti@iiit.ac.in

Abstract. The paper describes an annotation scheme for English based on Panini's concept of karakas. We describe how the scheme handles certain constructions in English. By extending the karaka scheme for a fixed word order language, we hope to bring out its advantages as a concept that incorporates some 'local semantics'. Our comparison with PTB-II and PropBank brings out its intermediary status between a morpho-syntactic and semantic level. Further work can show how this could benefit tasks like semantic role labeling and automatic conversion of existing English treebanks into this scheme.

1 Introduction

Beginning with the Penn treebank [14], treebank annotation has remained an important research area in CL and NLP. The PTB itself has become richer by incorporating various facets of language phenomenon over the basic phrase structure syntactic representation. Some of these include addition of grammatical relations (PTB-II, [15], [14]), predicate argument structure (PropBank [11]), and immediate discourse structure (PDTB [16]). Treebanks in other languages have continued to enrich this research initiative. For morphologically rich languages like Czech, one major effort has been the Prague Dependency Treebank [8], which has used a dependency based formalism. The Hyderabad dependency treebank- HyDT [1] for Hindi also follows the dependency based approach. In this paper we elaborate & extend the karaka based annotation scheme used in HyDT to English. We also compare some of our tags with similar tags in other well known schemes. As we will see from the examples discussed in the paper, karaka relations capture some level of 'local semantics'. As Rambow et al. [19] state, "local semantic labels are relevant to the verb meaning in question, while global semantic labels are relevant across different verbs and verb meanings". Previous work [18] has used an annotation scheme based on a dependency structure for English but our scheme differs considerably.

The paper is arranged as follows; in Section 2 we discuss the concept of karaka relations. Section 3 describes the data used for annotation. In Section 4 we explain the tagset used. We show how some English constructions are handled in the scheme in Section 5. Section 6 compares our work with a dependency version of Penn Treebank as well as with PropBank. We discuss some related issues in Section 7.

A. Gelbukh (Ed.): CICLing 2009, LNCS 5449, pp. 41–52, 2009.

2 Karaka Relations

The annotation scheme carries out the analysis of each sentence taking into consideration the verb as the central, binding element of the sentence. Sanskrit grammarians like Panini and later Tesnière [22] have used this idea in their grammars. The concept of syntactic valency, where a verb plays the central role has been applied to English before [9].

In this scheme, the verb's requirements for its arguments are the starting point of the analysis. Both arguments and adjuncts are annotated, taking into consideration the verb meaning. Their relationship with the verb is described using relations that we call **karaka** relations. This is a term borrowed from Sanskrit grammar to describe the way in which *arguments participate in the action described by the verb*. We claim that the notion of karaka will incorporate the elements of the local semantics of a verb in a sentence, while also taking cues from the surface level morpho-syntactic information.

For example, *karta* or k1 is a relation that describes an argument that is *most central to the action described by the verb*. The discovery procedure for a karaka like k1 uses such a semantic definition as well as certain morpho-syntactic information. We will discuss the discovery procedure of k1 to give a sense of the type of analysis we have carried out. Some of these tests were created after a pilot annotation of some English sentences.

In a sentence with a finite, transitive verb like ***John gave the flowers to Mary,*** John is a clear candidate for k1, as John is the locus of the activity of the particular verb in the sentence. Moreover, the verb agrees with John and occupies a position to the left of the verb. Both these are also important clues. But, the position of the argument is not always useful. In a sentence like ***To Mary, he gave the flowers; to Susan he gave nothing*** (example from [13], the position of the constituents will not help us. In that case, we will use the other tests of agreement and semantic relationship.

To elaborate further, in the following sentences :

i. The boy opened the lock.
ii. The key opened the lock.
iii. The lock opened. Example from [4]

The boy, the key and the lock will be annotated as k1. In case of ii and iii, the key and the lock are not actually the agents, but in the 'local semantics' of the sentence, i.e. the portion of the action described by the verb, they are the central participants. Hence, these relations differ from the broader 'global' semantic relations of Agent, Patient, Goal etc. To some extent, the notion of k1 corresponds with that of Subject, but there are some important differences. We have discussed these in Section 6. The differences are also apparent in the way subjects for Passives and Expletive sentences are handled (see section 5).

Note that the discovery procedure for k1 in the case of a passive sentence will also take into account prepositional information such as the preposition 'by'. Similarly, we can take into consideration prepositions like 'with' for annotating the relation k3 – instrument essential for the action to take place. (For example, a sentence like *John*

cut the fruit ***with*** *a knife*). The prepositions are additional clues for the discovery of karaka relations along with the semantic information.

3 Data

The corpus used for annotation consisted of 500 POS tagged sentences from the Wall Street Journal section of the Penn Treebank. The corpus was first converted to the Shakti Standard Format (SSF) [3].

Each sentence was manually chunked and then annotated for dependency relations. While chunking, we assumed that a chunk was a minimal, non-recursive structure consisting of correlated groups of words [2][1].

Karaka relations were marked among chunk heads rather than among each word, as the emphasis was on showing the right modifier-modified relationship. In addition to these, we also annotated verbal nodes with feature structure information. For instance, in order to handle cases with expletive 'it' ('It is raining') we add <stype=expletive__it>[2].

As the task was a preliminary one, a total of two annotators worked on the data. The corpus was small and as the annotators worked on a separate set of sentences, no comment can be made about inter-annotator agreement at this stage.

4 Tagset

We will elaborate on the tagset used in this section. (Fig. 1) shows the hierarchical nature of the tagset. 'Advmod', 'nmod', 'vmod' and 'jjmod' correspond to the adverb modifier, noun modifier, verb modifier and adjective modifier respectively. Below the

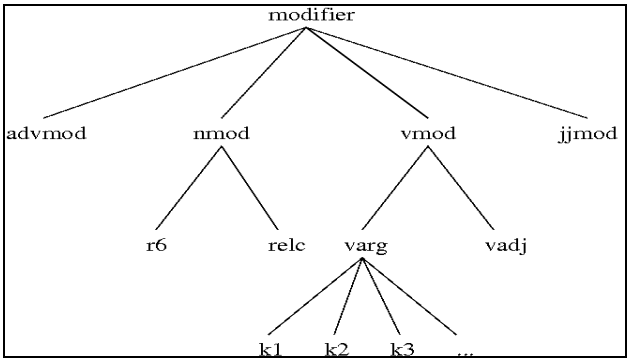

Fig. 1. Hierarchical tagset

[1] This particular chunk definition was used in order to facilitate an English-Hindi machine translation task.

[2] The double underscore '__' is read as 'of the type', and provides a more fine grained classification of the element to its immediate left. So, 'stype__expletive__it' would be read as, *sentence type 'of the type' expletive ' of the type' it.*

noun modifier, we have the noun dependencies of r6 (possession) and relc (relative clause). Similarly, below the verb modifier we have the verb arguments, which are the karaka labels, k1, k2 and so on. This can continue to be expanded into more fine grained labels based on the need. Hence, a relation like k1 can be further divided into different types.

Based on this hierarchy, we list the important noun and verb modifiers (nmod and vmod). (The complete list of tags may be found here[3]). In addition to the relations based on an expansion of the nmod and vmod nodes, we also have the tags 'fragof' and 'ccof'. These do not represent the kind of modification that is vmod or nmod, but show other kind of relations among chunks. For examples, see section 4.3.

The tagset is relatively small (currently 24 tags). Below, the SSF format shows the actual annotation format, with dependency relations and feature structures marked at the chunk level. The SSF representation shows four columns for node index, token (and chunk boundaries), tag and feature structure respectively.

```
<Sentence id="1">
0     ((        SSF
1     ((        VG      <drel=fragof:1>
1.1   Did       VBD
      ))
2     ((        NP      <drel=k1:1>
2.1   Rama      NNP
      ))
3     ((        VG      <name=1/stype=interrogative__yes-no>
3.1   eat       VB
      ))
4     ((        NP      <drel=k2:1>
4.1   the       DT
4.2   banana    NN
4.3   ?         ?
      ))
      ))
</Sentence>
```

The corresponding dependency tree can be seen in Section 4.1, (Fig. 6) The node indexed with 2 for instance shows the chunk boundary of NP followed by the chunk label and the feature structure containing the karaka label (k1) and its head, which is VG (node 3), marked as <name=1>. Using information from the edge label (karaka or others), dependency attachment, feature structure and the word order retained in the format above, the analysis of a sentence is carried out.

4.1 Verb Modifiers

The Sanskrit grammar system described by Panini assigns karakas to verbal arguments based on the relationship they have with the verb. In the annotation effort

[3] http://sites.google.com/site/deptagset/Home?previewAsViewer=1

described here, we have followed the way in which karakas have been defined in Paninian grammar. He classifies six karakas according to the way in which they participate in the action of the verb. These may be listed as follows, with the approximate translations from the sutras that mention them [21]:

k1: *karta*: central to the action of the verb
k2: *karma*: the one most desired by the karta
k3: *karana*: instrument which is essential for the action to take place
k4: *sampradaan*: recipient of the action
k5: *apaadaan*: movement away from a source
k7: *adhikarana*: location of the action

One of the peculiarities of the karaka system according to Panini shows that constructions like active and passive are the realizations of the same structure apart from certain morphological distinctions [12].

We follow the same principle to handle the case of passives in the annotation scheme (Fig. 3). While (Fig. 2) shows the analysis of an active sentence, the same dependency tree is drawn for the passive, only marking the verb's TAM (Tense, Aspect & Modality) as passive. The feature structure that marks the verb morphology as passive will indicate that the agreement and positional information in the tree is applicable to k2 and not k1 (see (Fig. 3), cf. Section 2).

The difference between the two constructions is lexical (and morphological) rather than syntactic in this framework. Such relocation of syntactic information into the lexicon is not unique; frameworks such as MTT [10] also make extensive use of the lexicon to account for various linguistic phenomena.

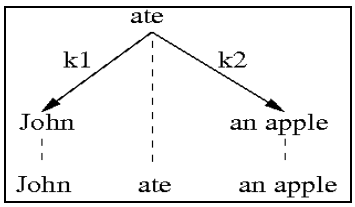

Fig. 2. An active sentence

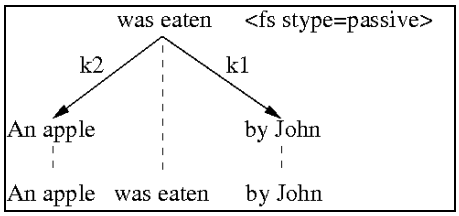

Fig. 3. A passive sentence

4.2 Noun Modifiers

Noun modifiers consist of noun-noun relations such as the possessive relation **r6.** It will hold between two noun chunks. For example, in 'The book of John' the head will be [The book] and [of John] will have a relation of possession (**r6**) with it. Those relative clauses that modify nouns will be marked as **nmod__relc.** (Fig. 4). The figure clearly shows the verb 'joined' as the head of the relative clause and the relative pronoun 'who' which is coreferenced with 'students'.

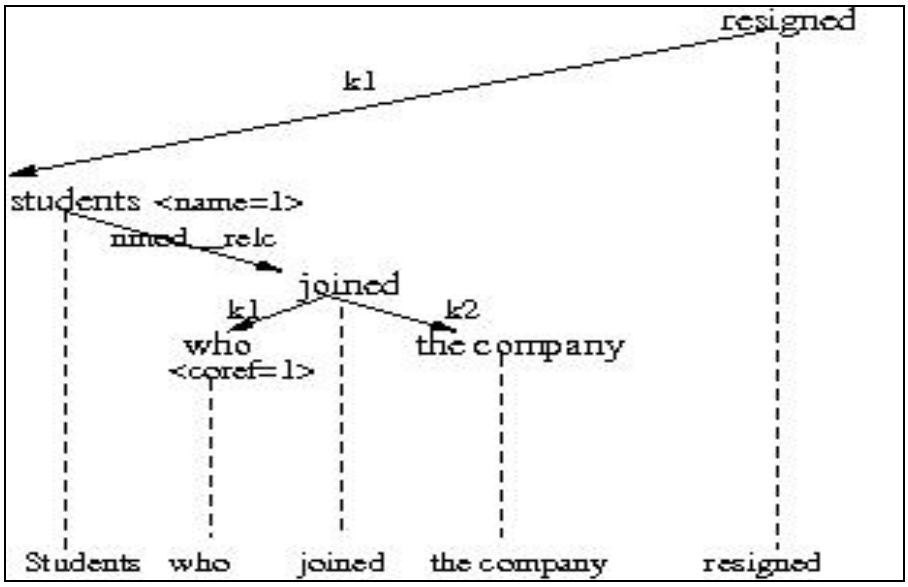

Fig. 4. Relative clause

4.3 Other Labels

In addition to the karaka labels of the hierarchy, we handle co-ordination using the label ccof 'conjunct of'. (Fig. 5) below gives an example for coordination:

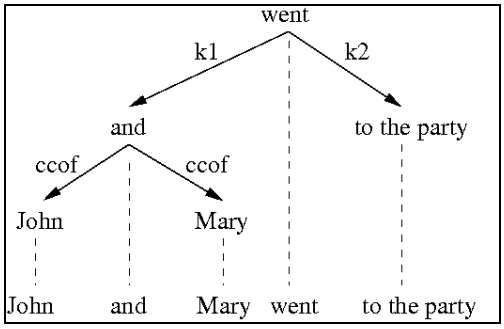

Fig. 5. Co-ordination

Note that in this case, 'and' which is a functional element acts as the head. This label can be used for co-ordination relations among constituents of any kind-noun, verb or adjective chunks. It should also be noted here that unlike the relations discussed earlier, 'ccof' is not a pure dependency label. The problem of representing coordination in the dependency framework is well known, different schemes follow different strategies. In this respect our handling of coordination is close to Prague dependency framework [8].

5 Some Constructions

In this section, we will elaborate on our annotation scheme with the help of some syntactic constructions in English.

5.1 Yes-No Questions

In English, interrogative sentences can be of yes-no type or use a wh-element. In both cases, we treat the displaced element without the use of traces. Instead, the moved constituent is analyzed *in situ*. In the case of yes-no questions, (Fig. 6) shows the dependency tree. We add the information that the sentence is a yes-no type of interrogative sentence. The moved TAM marker is given the label 'fragof'[4] to show that it belongs to the verb chunk that is its head. Finally, we mark the remaining arguments of the verb with karaka relations.

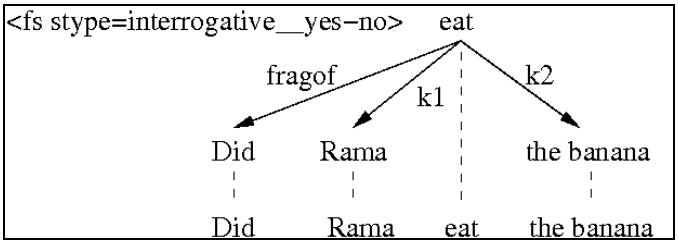

Fig. 6. Yes-no questions

5.2 Control Verbs

For English, the control verbs such as *promise* or *persuade* are not analyzed as cases with an empty PRO. Instead, the analysis shows a difference in the verb semantics of promise and persuade, which again amounts to making the lexicon richer. Traditionally, for an object-control verb like persuade, the object of persuade is coindexed with the missing subject of the subordinate clause.

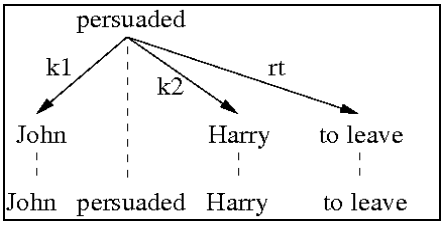

Fig. 7. Object control verb

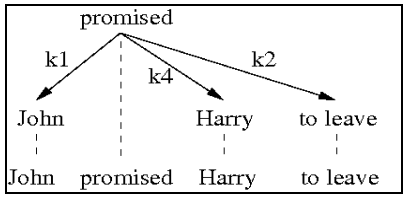

Fig. 8. Subject control verb

[4] *Fragment of.*

In (Fig. 7), the tree does not show a missing element but analyses the verb semantics of persuade differently from the semantics of a verb like promise (a subject-control verb) in (Fig. 8) Persuade is shown to take a karta (k1), a karma (k2) and tadarthya (rt or purpose) labels. Promise on the other hand takes karta(k1), karma(k2) and sampradaan (recipient of the action -k4) labels.

5.3 Expletive Subjects

In (Fig. 9), 'It' is a dummy element in the sentence to fill the empty subject position. We mark it with a special relation 'dummy', which reflects the fact that 'It' is semantically vacuous. We also add the information about the expletive construction to the feature structure of the head. The semantically vacuous 'It' will fail the test for k1 although it is in the subject position and agrees with the verb (Section 2 lists some of the criteria to test for k1).

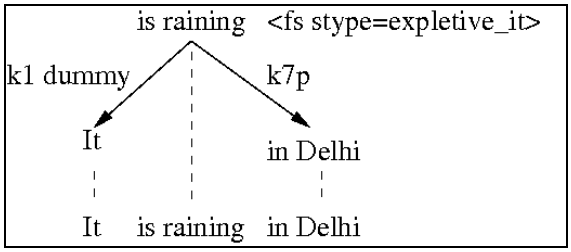

Fig. 9. Expletive sentence

5.4 Subordinate Clauses

Verbs such as *want* that take subordinate clauses can be represented where the subordinate clause is related with the relation k2 'karma'. This analysis is a direct consequence of the semantics of the verb and the way 'k2' is defined (Section 4.1). In (Fig. 10) for example, 'want' takes 'to leave' as its immediate child with a 'k2' relation and 'Harry' is shown attached to 'to leave 'with a relation 'k1'. It is easy to see that (Fig. 10) reflects the predicate argument of 'want' and 'leave'.

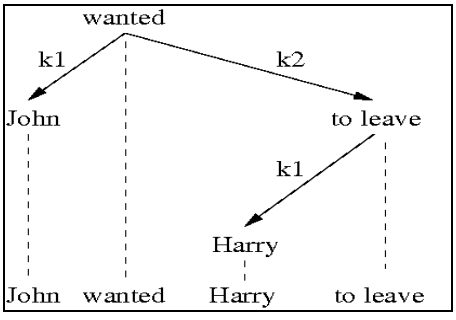

Fig. 10. Subordinate clause

6 Comparison with Other Schemes

As the data used of annotation has been taken from the WSJ[5] section of the Penn Treebank, it is possible to compare our labels with PropBank labels and labels taken from a converted PTB-II treebank. It has been automatically converted into dependency trees [23] and henceforth for convenience, we will refer to it as PennDep. In this section we sketch a preliminary outline about how the karaka labels relate to the syntactic labels in PennDep and the semantic labels of PropBank. We only show mapping between some important labels, an exhaustive mapping analysis is out of the scope of this paper.

Mapping the karaka labels with PennDep labels is straight forward: if the incoming edge of a node in the PennDep tree is marked with some label we find its correspondence in our dependency tree. While mapping NP-SUBJ[6] to k1, we found that 10% of the total NP-SUBJs do not map to k1. This 10% is distributed over **Expletives**, **Passive constructions**, **Coordinating conjunctions** and **Non-finite clauses**.

As mentioned earlier, the notion of k1 though syntactically grounded, also entails some semantics. In the case of expletives which are syntactic subjects but are semantically vacuous there is a mismatch (also see, Section 5.3). We see a mismatch for coordinating conjunctions too, the subtree representing conjunction is rooted at the conjunct (see Section 4.3), this is not how conjuncts are represented in PennDep. In PennDep the conjunct becomes the child of the right conjoined element. Unlike PennDep where the object of an active sentence appears as a subject in the passive counterpart, the karaka labels in active-passive sentence do not vary; as mentioned earlier, the scheme looks at a passive construction as a realization of an underlying structure with lexical/morphological variation rather than syntactic. Such asymmetries between k1 and NP-SUBJ mapping point to the fact that the notion of the k1 karaka relation entails a level of semantics which is absent in NP-SUBJ.

Comparison between karaka labels and Propbank labels can be done at two distinct, though related, levels. One is definitional, where we can compare the assumptions which come into play while using terms such as Arg0, Arg1 etc. on the one hand and k1, k2, etc. on the other. The second level would be practical i.e. the comparison of the actual annotation practice and deciding the mapping criterion.

Unlike the mapping between karaka labels and PennDep labels, where simple label comparison of a node sufficed, the mapping in this case would be between a span of text (PropBank annotation) and a subtree (of a dependency tree). Different strategies can be arrived at while deciding the configuration of this subtree. For the purpose of this paper we follow an approach similar to DepPropBank 1 as described in [6] and compare it with the actual PropBank annotation spans. Hence, when mapping, k1 with ARG0, we see if the text span annotated as ARG0 is contained in the subtree rooted at the node which has a k1 relation with the predicate. The map is said to be valid only if all the lexical elements in the dependency subtree appear in the argument span.

The PropBank annotation marks the predicate-argument onto the syntactically parsed, or treebanked, structures. It aims to provide consistent argument labels across

[5] Wall Street Journal.
[6] Noun Phrase Subject.

different syntactic realizations of the same verb. These arguments are labeled as Arg0, Arg1 etc. The annotation also marks modifiers of the verb such as manner (MNR), locative (LOC) etc.

Arg0 arguments are the arguments which cause the action denoted by the verb, either as agents or as experiencers, for instance the arguments of stative verbs such as *love*, *hate*, *fear* etc. Arg1 arguments, on the other hand, are those that change due to external causation, as well as other types of 'patient'-like arguments [5].

It turns out that the karaka labels are very similar to the PropBank labels in this respect. Going by the definition of ARG0, it maps to k1 for most cases. However, k1 also maps to ARG1 in the case of unaccusative verbs. The distribution of k1 across various labels in PropBank is as follows:

(a) k1 → ARG0: ~59%
(b) k1 → ARG1: ~19%
(c) No map: ~20%

We have already observed that k1 will map with ARG1 in the case of unaccusatives. The statistics show ~20% cases where k1 did not map to any PropBank label. Looking at such instances, we found that this occurs mainly due to our chunking guidelines and the mapping strategy used. Some of the chunking decisions are related to **prepositions**, **negation**, **punctuations coordinating conjuncts**, **participle constructions** etc.

According to the chunking guidelines, a preposition appears as part of the chunk, eg. *for* in [For Mr. Winston Smith]; in PropBank only "Mr. Winston Smith" appears as an argument, say ARG0, of a predicate. Mapping the PropBank text span with a subtree in our dependency tree will therefore need a more refined strategy. In fact, [6] mention other mapping versions in their paper. Along with handling the effect of our chunking guidelines, we will have to consider other strategies to reduce this asymmetry. Considering that this problem arises mainly due to the difference in the guideline decisions and can be resolved, we can see that the mapping from karaka labels to PropBank labels can be achieved systematically using a controlled strategy.

Preliminary statistics and observations from the mappings described in this section show that the proposed scheme lies between the syntactic level of the PennDep and the semantic level of the PropBank.

7 Discussion

The comparison effectively brings out some of the properties of the karaka relations that we are annotating on the corpus. Such a comparative study also becomes essential from an interoperability perspective. Mapping various relations in the two schemes helps in bringing out important trends and issues, which prove to be pertinent while automatically converting a treebank from one scheme to other.

We think that the karaka based tagset will give considerable leverage to tasks such as semantic role labeling. We saw in Section 6 that mapping karaka labels with PropBank labels and PennDep labels show consistent pattern.

It has been shown that syntactic parsing helps in identifying semantic relations [7]. Similarly, the karaka based dependency trees proposed in the scheme should help in getting to global semantic relations.

Another characteristic of the scheme is that it has a hierarchical tagset. This means that many relations are left under-specified at the present stage. These relations are typically those which do not concern the argument structure of the verb; for example, noun-noun, participle-verb, and adverb-verb relations. One needs under-specification mainly due to practical concerns. Parsing experiments with very fine-grained labeled treebanks have shown that learning such labels is not always easy [17]. To maximize the overall performance of the parser different levels of granularity can be tried out.

There are a number of issues which still need to be worked out, some obvious phenomena not described in Section 5 are VP ellipsis, wh-movement and raising verb construction like 'seem'. Also, discourse level information needs to be captured using a richer feature structure. In terms of comparison with other annotation schemes for English, we need to carry out experiments with a larger corpus to understand its performance with respect to NLP tasks like semantic role labeling.

References

1. Begum, R., Husain, S., Dhwaj, A., Sharma, D.M., Bai, L., Sangal, R.: Dependency annotation scheme for Indian languages. In: Proceedings of IJCNLP 2008 (2008)
2. Bharati, A., Sangal, R., Sharma, D.M., Bai, L.: AnnCorra: Annotating Corpora Guidelines For POS And Chunk Annotation For Indian Languages. Technical Report, Language Technologies Research Centre IIIT, Hyderabad (2006)
3. Bharati, A., Sangal, R., Sharma, D.M.: Shakti Analyser: SSF Representation (2005), http://shiva.iiit.ac.in/SPSAL2007/ssf-analysis-representation.pdf
4. Bharati, A., Chaitanya, V., Sangal, R.: Natural Language Processing: A Paninian Perspective. Prentice-Hall of India, New Delhi (1995), http://ltrc.iiit.ac.in/downloads/nlpbook/nlp-panini.pdf
5. Babko-Malaya, O.: PropBank Annotation Guidelindes (2005), http://verbs.colorado.edu/~mpalmer/projects/ace/PBguidelines.pdf
6. Ekeklint, S., Nivre, J.: A Dependency-Based Conversion of PropBank. In: Proceedings of FRAME 2007: Building Frame Semantics Resources for Scandinavian and Baltic Languages, pp. 19–25 (2007)
7. Gildea, D., Palmer, M.: The Necessity of Parsing for Predicate Argument Recognition. In: Proceedings of ACL 2002 (2002)
8. Hajicova, E.: Prague Dependency Treebank: From Analytic to Tectogrammatical Annotation. In: Proc. TSD 1998 (1998)
9. Herbst, T.: English Valency Structures - A first sketch. Technical report EESE 2/99 (1999)
10. Kahane, S.: The Meaning-Text Theory. In: Dependency and Valency. An International Handbook on Contemporary Research. De Gruyter, Berlin (2003)
11. Kingsbury, P., Palmer, M.: From Treebank to PropBank. In: Proceedings of the 3rd LREC, Las Palmas, Canary Islands, Spain (2002)
12. Kiparsky., P.: On the Architecture of Panini's grammar. In: Three lectures delivered at the Hyderabad Conference on the Architecture of Grammar (2002), http://www.stanford.edu/~kiparsky/Papers/hyderabad.pdf

13. Kroeger., P.: Analyzing Syntax: A lexical functional approach. Cambridge University Press, Cambridge (2004)
14. Marcus, M., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: The Penn Treebank. In: Computational Linguistics (1993)
15. Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn treebank: Annotating predicate argument structure. In: Proceedings of the ARPA Human Language Technology Workshop (1994)
16. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. In: Proceedings of the 6th LREC (2008)
17. Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering 13(2), 95–135 (2007)
18. Rambow, O., Creswell, C., Szekely, R., Taber, H., Walker, M.: A dependency treebank for English. In: Proceedings of the 3rd LREC, Las Palmas, Gran Canaria, Spain (2002)
19. Rambow, O., Dorr, B., Kucerova, I., Palmer, M.: Automatically Deriving Tectogrammatical Labels from other resources- A comparison of Semantic labels across frameworks. The Prague Bulletin of Mathematical Linguistics 79-80, 23–35 (2003)
20. Sgall, P., Hajicova, E., Panevova, J.: The meaning of the sentence and its semantic and pragmatic aspects. Reidel, Dordrecht (1986)
21. Subrahmanyam, P.S.: Pa: ninian Linguistics, Tokyo, Japan: Inst. for the Study of Languages and Cultures of Asia and Africa, Tokyo University of Foreign Studies (1999)
22. Tesnière, L.: Eléments de Syntaxe Structurale. Klincksiek, Paris (1959)
23. Yamada, H., Matsumoto, Y.: Statistical dependency analysis with support vector machines. In: Proceedings of IWPT (2003)