Chapter 1

Introduction

Language resources form a crucial component in Natural Language Processing (NLP). “Language resources are machine-readable data sets and language descriptions” [60] which “are important in the developing, improving, and evaluating different natural language processing algorithms and systems” [61, 60]. Language resources not only contribute towards advances in research in the area of natural language processing but are also useful for research in linguistics. Lexica, text and speech corpora, grammar rules, terminology databases, tagsets, speech collections and processors, tokenizers, and annotated corpora, etc., are some of the examples of language resources. Lack of adequate language resources hampers the development of different natural language applications for resource poor languages. Creation of language resources, however, is a costly affair both in terms of time and effort and also monetarily. Often, it is also labour intensive. Thus, advanced NLP research can only be carried out for languages which have adequate language resources. This also leads to development of high quality NLP applications.

One of the important language resources is annotated corpora. “Annotated corpora are large bodies of text with linguistically-informative mark up” [65] and “they provide training material for research on natural language processing algorithms and serve as a gold standard for evaluating the performance of such tools” [65]. Recently, corpora that have been syntactically annotated have been created for various languages. Syntactically annotated corpora are a reliable base for resource development for further annotation such as semantic and discourse annotation. Automatic extraction of linguistic knowledge such as grammar and lexicon extraction can be done from it. Automatic parsing tools can be created. Annotated corpora also form a base for navigation and search, i.e., query languages and support engines. Syntactically annotated corpora are referred to as treebanks as the sentences parsed manually are represented in tree structures. The term ‘treebank’ was coined by Geoffrey Leech [155].

In treebanks, linguistic information is annotated at various levels, i.e., at the word level, phrase level, clause level, and sentence level, hence forming a bank of linguistic trees. With the increase of more and more treebanks for various languages, there is also an increasing understanding of the importance and effect of linguistic theory for the layout of treebank annotation schemes [70]. “Over the years, a number of distinct schemes have been introduced for syntactic annotation” [130,161]. Various grammar formalisms are used for developing the schemes for syntactic annotation. Phrase structure based schemes and Dependency grammar based schemes form major approaches in this area.

I present my work on a pilot Treebank for Hindi (HyDT - Hyderabad Dependency Treebank) [21] in this thesis. I describe the annotation scheme [21], the theoretical basis for the scheme and the resources which I created for facilitating consistent annotation.

1.1 Motivation

Apart from their use for data driven parsers and other NLP tasks, “the need for treebanks as an empirical basis for research on the grammar of a language is well established [134]. While it is often stated that treebanks are useful for linguistic research as well as for language technology [133], linguistic research with treebanks seems under-represented in the literature” [134]. Thus, to summarize, the aim or need behind creation of treebanks is as follows:

1) For construction and evaluation of NLP tools, such as, word segmenters, part-of-speech taggers, chunkers, parsers, semantic role labellers, and named entity detectors, etc. Treebanks are also used in Discourse, Co-reference, and Event detection.

2) Parsing is important for natural language understanding. It analyses the structure of sentences [156] and it needs a grammar to be able to analyse real text accurately. For this, treebanks could be one of the sources of grammar extraction [56, 50, 156]. Parsers, trained on treebanks are more robust and successful for practical applications [185].

3) Treebanks are used as a repository for linguistic research. They are used in corpus-based linguistic research that is related to syntax. Recently, annotation in treebanks has gone beyond syntax, to semantic features like predicate-argument structures [185].

4) Treebanks are also used in pedagogy, i.e., it is used in language teaching and in the teaching of linguistic theory.

Now a days, the term treebank is not just used for corpora having tree structures, but is also used for all types of grammatically analyzed corpora.

The creation of Penn Tree Bank [110] changed the NLP research scene significantly. It contributed to major advances in NLP research in general and English in particular. The other treebanks and language resources in English that contributed to major advances in NLP research are: Prague English Dependency Treebank, LinGO Redwoods Treebank [137], A Dependency Treebank for English [145], Wordnet [67], Verbnet [101, 100], Framenet [16], PropBank [139] etc.

If we move to the Indian scene, on the other hand, we find very few linguistic resources exist for Indian languages, even for the major Indian languages such as Hindi, Tamil, Telugu, Marathi, Bengali, etc. Multilinguality has survived in India through the ages. Several languages have rich literary history. Indian nation is a composite of several states which were formed on linguistic basis. Thus, each state chose one of its languages as its official language. In the advancing technological world of large body of information exchange and faster communication, NLP based applications are gaining fast grounds. This implies that languages which are not ready to be a part of this advancement will be left behind. Therefore, India needs major technological advances in the field of linguistics and language technologies for technologically supported official and administrative purposes. In this context, it becomes crucial to have
more linguistic resources for Indian languages. In the recent years, some POS, Chunk annotated corpora have been created for some major Indian languages.

1.2 Problem Statement and Scope

Creation of a treebank for Hindi is the focus of this thesis. A treebank, as mentioned above, is a hand annotated corpus for syntactic analysis of each sentence in the corpus. This assumes a grammatical model for annotating the corpus. For the annotation design of a treebank, there are at least two different levels that are to be differentiated [131]:

a) Annotation Scheme: It is the level of linguistic analysis that has some assumptions regarding the type of syntactic structure, a particular selection of linguistic categories, and guidelines for the annotation of specific linguistic phenomena.

b) Encoding: At this level a decision is taken about the representation of the annotation, i.e., whether the representation should be done using a special markup language or ordinary text, whether it should be kept in one file or several files, etc.

The selection of annotation scheme for an extensive treebank is affected by many different elements. The major one is the relation to linguistic theory, i.e., which theoretical framework has to be selected. There are other things, such as the typical grammatical features of the language that is being worked on, and the tradition of descriptive grammar that is there for this particular language [131].

The two major grammar formalisms that are used widely which represent the syntactic structure of a sentence are [72]:

(i) Constituency (or phrase structure) and

(ii) Dependency

In these grammars, the structure of a sentence is represented using trees, though the meaning of the nodes and links is distinct in the tree. In the phrase structure grammar, the nodes of the tree are text spans and the links denote inclusion relation. In dependency tree, the nodes of the tree are single words. A dependency relation gets created between a pair of words. One of the words is the head word (governing) and the other (subordinate word) is its dependent, i.e., dependent of the head word [72]. Penn English Treebank, Penn Chinese Treebank, and Penn Korean Treebank are annotated in Phrase structure grammar based model. Apart from Phrase structure grammar based models, there are treebanks, such as Prague Dependency Tree Bank (PDT) [81] and Turin University Treebank [46] which have followed dependency grammar [171] based models.
Since Hindi is an Indian language and has relatively free word order [64, 36], dependency grammar formalism is very well suited for it. In such languages, because of their rich morphology, there is more freedom in word order for expressing syntactic functions [171, 95]. Most of the languages including Indian languages have relatively free word order. It is necessary to have an appropriate computational grammar formalism for free word order languages for two reasons [36]:

I. An appropriately planned formalism is effective to a great extent as it can use basic sources of information directly.

II. Formalism of such type will be linguistically more elegant and convincing. The grammar will be more economical and simple to write as it relates to primary sources of information.

Paninian Grammatical Framework is one such formalism that has been effectively used for Indian languages [36]. Paninian grammar is a dependency based grammar [99, 162]. It is inspired by Sanskrit language which is an inflectionally rich language. Paninian grammar provides a karaka based (dependency relations) analysis framework for a sentence. The Paninian grammar treats a sentence as a series of modifier-modified relations where a sentence is supposed to have a primary modified (the root of the dependency tree) which is the main verb of the sentence. In Paninian grammar, Hindi postposition/case markers are referred to as vibhaktis. vibhakti denotes the case markings on the nouns and the TAM (tense, aspect and modality) of the verbs. vibhaktis play a key role in indicating semantic relationships.

The karaka relations are syntactico-semantic in nature [36]. The grammar provides a means for syntactic analysis as well as includes semantic information. Here the main verb is the central binding element of the sentence.

We chose to use Computational Paninian Grammar (CPG) [36] model for the Hindi treebank. Developing a treebank for Hindi was first envisaged in [34]. Since Hindi is a free word order language and we chose CPG as the grammatical framework for annotating the dependency structures for syntactic analysis of a sentence, the first question that we were faced with was:

(a) Whether Computational Paninian Grammar (CPG) would be adequate for analyzing various grammatical constructions of Hindi?

Although it was easy to arrive at (a) above, it immediately posed another question before us:

(b) How do we go about collecting/identifying various Hindi constructions?

The third question that came up before us was:
Whether analysis of various constructions was enough to understand the linguistic issues for the annotation task or some additional resources are required?

## 1.3 Major Contributions of this Research Work

For addressing (a), (b) and (c) we decided to annotate some Hindi corpus with an existing version of the scheme and guidelines [34]. We chose a small part of the Hindi corpus developed at CIIL (Central Institute of Indian Languages, Mysore, India) for this purpose. The part that we chose largely, turned out to be stories by Premchand. Around 2230 (no. of unique tokens = 34354) sentences were taken from these stories and were annotated. Through the process of dependency annotation we came across issues in: (a) actual annotation of naturally written sentences in a corpus and (b) certain Hindi constructions. We also realised that stories did not make the annotation task easy. However, on the other hand, we could resolve several linguistic issues related to various Hindi constructions. The process involved intense discussions interspersed with actual annotation of the corpus. The Guidelines were simultaneously updated for further annotation.

I started the process by taking 600 sentences out of the 2230 from CIIL corpus and annotated them with Computational Paninian Grammar (CPG) analysis. The process was repeated 2-3 times to check for the errors and to check if any of the sentences were left unannotated. The sentences which had unresolved issues were left unannotated initially. The issues were then discussed and once resolved satisfactorily, the sentences were revisited and were annotated according to the decision taken. Cross-validation of the data was also done. The annotation process went in parallel with the revision/modification of the guidelines. Through the process of pilot annotations and discussions, the guidelines [34] were substantially enhanced. The remaining set of sentences was also annotated similarly. Experiments to study inter-annotator agreement were conducted. After analysing the differences in the inter-annotator agreement, we found that these differences were due to:

a. The guidelines lacked descriptions of some cases and there was not much clarity, and

b. The annotation errors.

The above problems were addressed and the guidelines were modified. This phase of scheme development and guidelines creation was relatively slow. Once the guidelines with a more comprehensive scheme were ready, the task of annotation improved in quantity and quality. The total annotated data is 2230 sentences. Out of this, a set of 600 sentences was used in the initial phase. This Hindi Treebank (HyDT - Hyderabad Dependency Treebank) [21] was then released for a shared task on Indian Language (IL) parsing in ICON-2009 [86] as HyDT-Hindi\(^2\) (Hyderabad Dependency Treebank for Hindi) [21].

Thus, the scheme and guidelines for treebank annotation for Hindi were given a final shape which was subsequently used for a multi-layered and multi-representational treebank for Hindi and Urdu [42, 188]. This multi-layered and multi-representational treebank for Hindi [42, 188] was then released for the

\(^2\) The Treebank was developed at LTRC, IIIT, Hyderabad, India.
tools’ contest on Indian Language (IL) parsing in ICON 2010 [85]. Apart from Hindi, the scheme finalized through this process, was subsequently used for developing pilot treebanks for Telugu [182] and Bangla. Telugu and Bangla treebanks were named as HyDT-Telugu and HyDT-Bangla respectively. The representation format in all the treebanks, including the ongoing project on multi-layered treebank for Hindi and Urdu [42, 188], is SSF (Shakti standard Format) [32]. POS and chunk information is marked using a set of guidelines [33].

The pilot Hindi dependency Treebank (HyDT-Hyderabad Dependency Treebank) is the result of team work. My contribution in this is that I annotated the Hindi corpus (2230 sentences) with the initial version of the scheme and guidelines. Through the process of dependency annotation I came across several issues related to various Hindi constructions such as Causatives, Complex Predicates, Relative Clauses, etc. I did the analysis of these Hindi constructions, i.e., studied the syntax and semantics of these Hindi constructions in literature and analysed it from Paninian perspective. I worked on the version next to the initial version of the scheme and guidelines, and improved it. So the end product of this process is the pilot Hyderabad Dependency Treebank (HyDT) for Hindi. On the other hand, work done in collaboration with others is that I discussed the problem issues related to various Hindi constructions with the group. We developed a scheme for annotating these constructions based on Computational Paninian Grammar (CPG). The work on final guidelines was done by others.

Thus, the following concrete resources were created during the process:

1) **Dependency Analysis of Various Hindi Constructions:** For developing the scheme and the guidelines, we looked at various Hindi constructions and checked whether the chosen grammar formalism (Paninian dependency relations) could handle them all [21]. The analysis of the Hindi constructions gave us a better understanding of both Hindi data as well as the Paninian grammar. Hindi (like other languages) has various constructions such as Complex Predicates, Discontiguous elements, Ellipsis, Dative Subjects, etc., which makes the treebank annotation process more challenging.

2) **Creation of guidelines for dependency annotation:** Developing guidelines takes time. It is a necessary document for any high quality resource creation. The guidelines cover, apart from the scheme, the diagnostics and how to annotate various constructions in the concerned language. Thus, it involves a thorough study of various linguistic phenomena in the concerned language and coming to taking decisions in favour of the best possible solution which is both linguistically correct and computationally effective. Thus, the creation of the guidelines was a major task. The task of guidelines creation was interspersed with the data annotation, which included revision of an earlier draft version of the guidelines, throughout the annotation process of 2230 sentences.

3) **Creation of a Pilot Hindi Dependency Treebank (2230 sentences):** HyDT (Hyderabad Dependency Treebank for Hindi) [21] for Hindi aims to fully annotate Hindi corpus with syntactic (dependency) relations between the heads of the chunks. It was the first attempt of its kind to develop a treebank for an Indian language. HyDT is used in shared task on Indian Language (IL) parsing in ICON-2009 [86]. HyDT is used in the Hindi parser [84, 25]. Empirical results of some experiments done on this treebank are also reported [21].
4) Developing Verb Frames for Hindi (Verb entries created for 687 Hindi verbs): A verb frame captures various syntactic distributions in which a verb can be expected to occur in a language. A verb frame, in other words, captures the argument structure of a verb for various senses (Hindi in our case). I have also classified the Hindi verbs based on their argument structure. The main objective of this work is to create a linguistic resource which will prove to be indispensable for various NLP applications. We also hope this resource to help us better understand Hindi verbs. We motivate the basic verb argument structure using relations as introduced by Panini. We show the methodology used in preparing these frames and the criteria followed for classifying Hindi verbs [22].

During the process, I also looked at the following phenomena more extensively:

a. Complex Predicates: Conjunct verbs are highly productive and frequent phenomena in most Indian languages, especially so in Hindi. Since conjunct verbs, which are multi-word expressions can involve a noun and a verb, the decision about when a noun-verb sequence is a conjunct verb and when it is not, is a highly challenging task. They have been exhaustively studied by the researchers in the past. Some notable efforts in this direction have been Greaves [73], Kellogg [96], Mohanan [123], Butt [52], etc. Identifying a conjunct verb is a major task and then connecting the noun of the conjunct verb with its verb to show that they form a single unit is again a major task [19]. A novel solution for this problem is given in the conjunct verb section (Section 7.2).

b. Causatives: The section on causatives (Section 7.1) introduces the work on Hindi causative verbs: their classification, a linguistic model for their classification and their verb frames. The main objective of this work is to come up with a classification of the Hindi causative verbs. In the classification, we show how different types of Hindi verbs have different types of causative forms. Here, Hindi verbs have been classified into 6 types based on their causativization behaviour [20]. We present the morphology, semantics and syntax of the causative verbs. It will be a linguistic resource for Hindi causative verbs which can be used in various NLP applications. This resource enriches the already available linguistic resource on Hindi verb frames [22]. This resource will also be helpful in getting proper insight into Hindi verbs [20] and could complement and enrich the PropBank analysis.

1.4 Thesis Outline

This thesis is organized as follows: Chapter-2 provides information about various treebanks and the distinct approaches that have been followed for the representation of syntactic analyses in these treebanks. In Chapter-3, we describe the Grammatical Model followed in the HyDT (Hyderabad Dependency Treebank for Hindi) [21]. For creating a treebank, a Grammatical Model is needed for parsing the sentences in the treebank. In Chapter-4, we present the Annotation Scheme that we have followed for developing HyDT (Hyderabad Dependency Treebank for Hindi) which includes POS tagging, chunking, and dependency annotation, etc. In Chapter-5, we explain the various Hindi constructions that have been
problem cases as well as interesting from the point of view of their representation in the dependency structure. We have studied and analysed these Hindi constructions and have shown how Paninian dependency annotation scheme handles all these constructions very well [21]. In Chapter-6, we describe our work on *Verb Frames* which has the analysis of Hindi verbs and their argument structure. We have also classified the Hindi verbs based on their mandatory arguments. In this Chapter, we also present the comparison of *karaka* (dependency relations) relations with Propbank roles. In Chapter-7, we describe in detail about the *causative* verbs, their types, and their classification. In this chapter, we also discuss about *conjunct* verbs, diagnostics to identify them, and their automatic identification. In Chapter-8, we present some experiments that were conducted on a corpus of 2230 Hindi sentences that have been fully annotated. At the end, in Chapter-9, we give a short conclusion and future-work of this thesis.