सार्थक
A Generalized HPSG Parser for English and Hindi

D Sharma, K Vikram, Manav R Mital, A Mukerjee, A M Raina
Indian Institute of Technology, Kanpur,
Kanpur 208016,
Uttar Pradesh, India.
Email: {*deepaks,vikram,manavrm,amit,achla*}*@iitk.ac.in*

Abstract

This paper reports our work on **सार्थक**, a generalized syntactic parser, which can, in principle, be claimed to work for a multilingual input. Based on the HPSG framework, the parser operates on input strings from Hindi and English, two of the most structurally disparate languages used in India. The system is based on partial grammars of Hindi and English, which we have developed using the existing HPSG formalism. The parser yields multiple syntactic parses, which are then subjected to word sense disambiguation to arrive at an ordered list of sense combinations. Based on this, the system arrives at a set of parses in a decreasing order of appropriateness. This system is a part of a larger effort aimed at developing a unified semantics for Hindi and English discourses.

1 Introduction

Most of the available work on natural language parsing systems focuses on devices that operate on a single language to generate the parsed output. No serious effort has been made to devise generalized parsers that would work on more than one languages simultaneously, although some success on structurally related languages such as English and French, or English and German has been reported. This paper outlines our ongoing work on a generalized parser for Hindi and English, two structurally disparate languages used in the Indian multilingual context. The effort is a first step toward a natural language understanding system that would operate on typically bilingual/multilingual text inputs within limited story domains. The larger aim of the effort is to provide multilingual tools for animating textual stories ⟦Amitabha Mukerjee *et al.*2000⟧ in Hindi and English created by children and semi-literate adults.

सार्थक, the bilingual parser developed here is based on the HPSG framework, which integrates many of the crucial empirical and conceptual advances in linguistic theory. Whereas the grammatical description for English is drawn from the available work on this language ⟦Pollard and Sag1994⟧, a partial

HPSG grammar of Hindi has been developed here to provide the basis for the generalized parser. The bilingual parsing system works on an invariant set of principles that encompass both Hindi and English grammars. The parameters such as word order and morphology on which the two languages differ significantly are incorporated into the lexicon.

The output of the parser consists of an ordered set of all possible parses for each string. The multiple parses generated by the parser form the input to a word sense disambiguator, which arranges the output in a decreasing order of appropriateness. The word sense disambiguation device uses the ontology derived from the WordNet ⟦Fellbaum1998⟧

In its current phase of implementation, the parser operates on English and Hindi input strings, ranging from simple sentences to structurally complex embeddings of various kinds. The parsing methodology is fairly general and the parser can be developed further to work on input from a diverse set of languages without any fundamental changes in system design.

We begin with a brief description of the past work in the field (section 2) and the theoritical framework used in this study (section 3). Section 4 deals with the methodology, and section 5, the results comprising linguistic, computational and semantic analyses. The paper ends with a brief summary of the results and the conclusion in section 6.

2 Relation to Past Work

Considerable work in the area of parsing techniques has been reported in the literature ⟦Copestake and Flickinger2000, Carpenter and Penn1994⟧. But most of the parsers developed so far do not appear to be general enough to incorporate more than one language. Parsers such as the one used by ⟦Mukerjee *et al.*1999⟧ have had considerable success in parsing a single language, whereas LKB, originally developed for English, has been successfully extended to languages other than English . However, a serious limitation of both these kinds of parsing systems is that they generate an output consisting of multiple sets of parses even for the simplest strings, leaving the task of identifying the acceptable parses to the user. सार्थक is an advance on the existing parsers to the extent that it simultaneously operates on a bilingual input, and, that it generates a list of parses in decreasing order of appropriateness.

सार्थक provides us with an environment in which we can build databases describing grammars of many languages and also use it for syntactic tagging of input sentences. Such Parsers based on commonly employed grammatical formalisms like the ATN, TAG, CFG, and HPSG already exist but few of them claim to deal with both the syntactic and semantic issues of more than one language at the level that we intend to do.

As already stated, systems such as the LKB ⟦Copestake and Flickinger2000⟧ and LilFes or LILOG ⟦Erbach1991⟧ or even Anusaaraka ⟦Chaitanya *et al.*1997⟧ can be made to process input from more than one language. However, such systems have not been extensively tested on languages like Hindi which involve order independent unification. The LKB system is most commonly used with the LinGO ERG (English Rescource Grammar). Although LKB also incorporates Minimal Recursion Semantics ⟦Copestake *et al.*1995⟧ our treatment of semantics is deeper, and is directly linked to the WordNet [[Fellbaum1998]] which makes automatic acquisition of large amounts of semantic data possible.

In सार्थक, *semantics* provides a way of determining the appropriateness of a particular parse. This feature seems to be missing in most other freely available parsers, which leave the task of singling out the appropriate parse for any further applications to the user. Often this in itself is an enormous task. It has been claimed that in many of the cases the most appropriate parse depends on the sentential information itself, more than anything else. Thus it would be only reasonable to think that the process of singling out the most appropriate parse should be the job of the parser itself.

3 Theoretical Framework

सार्थक uses the HPSG framework for grammar representation and processing. Head-driven Phrase Structure Grammar [[Pollard and Sag1994]] provides a framework which is especially suitable both for purely linguistic research and for NLP applications. It rests on a meticulous and coherent formalism - the logic of typed feature structures - which is well-suited for computational applications, providing reversibility, declarativeness and the use of partial descriptions. Further, HPSG provides a formalism which is amenable to semantic treatment of linguistic input. Apart from English, which is the most comprehensively described, the HPSG framework has been adopted in studies of various other languages, e.g. German, Korean, Italian, Japanese, etc.

Parsing systems such as the XTAG [[XTAG1995]] which use the Tree Adjoining Grammar to increase the computational efficiency of parsing are significantly poorer in their semantic treatment [[Copestake and Flickinger2000]]. Even though some amount of semantics can always be incorporated in the syntactic lexicon itself, such a treatment can have limitations.

From a computational perspective it is known that Tree Adjoining Grammars are more efficient and efforts have been made to convert HPSG grammars to TAG kind of grammars [[Neumann and Flickinger1999]] to ensure faster processing. On the other hand, since HPSG has been found to be more amenable to treatment of semantics, work has also been done on converting TAG grammars to HPSG grammars. In fact, a few parsers claim to demonstrate higher efficiency with HPSG in comparison to TAG as their grammar representation. However, this conversion yields an 'HPSG-style grammar strongly equivalent to an original LTAG one prior to parsing'.

Also, emerging from the recent psycholinguistic literature is the observation that human linguistic sentence processing has a powerful lexical basis. Put simply, words are information-rich; hence, certain key words play a pivotal role in the processing of the clauses that contain them. This simple psycholinguistic insight is central to HPSG theory, since the notion of phrase structure is built around the concept of a lexical head and its subcategorization properties.

Although HPSG contains numerous feature sorts, all of them are not needed in the domain under study here. We are using an improvised subset of features mentioned in HPSG to suit our purpose. The feature list is extended to incorporate the peculiarities of Hindi. For instance, many nouns in Hindi, like कुत्ता, can be used either in their normal form or in the oblique form. Within the description field of nouns in the Hindi lexicon, we have added a feature which indicates the state of a noun. Both the entries for कुत्ता are as follows.

कुत्ता:कुत्ता N(N,-,S,M,1,V)
कुत्ते:कुत्ता N(N,-,S,M,0,V)

The fifth subfield in the features field is different for both these forms because of obliqueness. In the same spirit, gender and verb form agreement is essential in Hindi for any grammatical sentence in contrast with English. This is taken care of by inserting a new field in the description of verbs in the lexicon.

4 Methodology

4.1 Domain

At present, the work is restricted to a domain of about 400 words, which captures a typical family environment. There are enough concepts in the domain to generate rich and meaningful stories. The senses for all the words have been chosen using WordNet 1.7. The Devanag transliteration scheme has been used for representing the Hindi text.

4.2 Grammar Generation

Grammar generation involves building lexicons and descriptions of the languages under study. English is one of the most comprehensively studied languages within the HPSG framework, and several lexicons and descriptions of the language are already available. Hindi, however, remains relatively unexplored. Therefore we have developed a lexicon of Hindi and a partial grammar based on this lexicon ranging over simple sentences and complex sentences involving embeddings and adjuncts. Even for English, we have constructed a restricted lexicon of our own, primarily because the available lexicons are too large to be integrated into a semantic treatment that is crucial to our work. Nevertheless, all of this work on Hindi and English lexicons is scalable, the sizes of the lexicons being roughly of the order of the size of the domain. Extending the system to make it work on larger domains is a task that can be accomplished simply by adding entries to the existing lexicons. We have also built interfaces that makes it easy for a user, who is not habituated to the notations that we are using, to add entries to our database. Our system affords the user the freedom of using whatever notations and features he may choose, provided he uses them consistently throughout the lexicon. This has been done in order to achieve some flexibility in the system at the user end.

The entries in the lexicon are *Head-driven* in the spirit of HPSG, which implies that all the properties of a phrase are a function of the head of that phrase. We have also added constraints on the order of phrases/words in the sentence. There is also provision to represent the free word order phenomenon in Hindi which is more pronounced as compared to languages such as German.

The sample entry of the word `dog` in our lexicon could be something like this
`dog:dog` N(-,S,-,C) { D(S) | *J | ! }

The first field is the description of the word itself, in terms of its features. The second field is the set of features which can exist in the sub-category of this head word. The entry here means that the word `dog`, preceded necessarily by one and only one determiner(singular), and any number of adjectives would form a phrase with `dog` as the head. Similarly the word `dogs` would have
`dogs:dog` N(-,P,-,C) { ~D(P) | *J | ! }

The word following the colon is the root word of the one preceding the colon. This root word field is used during the semantic processing. Here one can see that the determiner is not compulsory. We have used certain such notations like '~' and '*' for the sake of convenience, so that the lexicon is easily manageable. '~' means that the feature occurring after it can occur either once or not at all and '*' means that the feature can occur any number of times. Again, the entry for `eats` would be
`eats:eat` V(P,S) { N(N,S,-,-) | ! | N(A,-,-,-) }

A sample entry of the word घर from the Hindi lexicon would look like as
घर:घर N(N,3,S,M,-,V) { ~D(S,M) | ~J(S,M) | ~k(S,M) | ! | }
The description of Hindi nouns are given with the help of six features as compared to four in English. Similarly, the subcategorization also differs at few parameters which are unique to each of the language.

Considering the entry for the word खाता which would be as
खाता:खा T(-,-,S,M,1,V) { / N(N,-,S,M,1,V) | [N(A,-,-,-,1,V) | a(V)] | ~i | ~p | ~s | ~l | ~Y \ | ! }

The notations '/' and '\' imply that all features that occur inside them can occur in any order in the sentence. Such a notation is necessary for a language like Hindi where word order is not as important as in English. The notation '[' and ']' mean the only one of all the features that occur inside these markers can occur in the sub-category. We could also have entries for expressions which might consist of more than one word. For instance the expression '`in front of`' in English and 'के लिये' in Hindi have an independent entry.

These lexicons are made to handle many complex constructs in the languages. In the lexicons, entries are kept such that sentences containing relative clauses, participial adjuncts, gerundival adjuncts etc. are also expected to get parsed. The system has been tested on a variety of complex sentences and results were as expected.

4.3 Parsing

Parsing involves searching for all possible phrase combinations which form a given sentence. In HPSG terms, it essentially boils down to finding the heads of all constituent phrases in a given sentence and the phrases that saturate the sub-categories of these heads. With sentences that are ambiguous all possible parses should be generated. Our system does not always assume a grammatically well-formed input; it is capable of recognizing sentences which are ungrammatical.

We apply a dynamic programming approach to the problem of parsing which essentially means that the main problem is broken down into sub problems. These sub problems are solved ensuring that the overlapping sub problems do not get re-solved. This procedure is called 'Chart Parsing' in which a chart is maintained which stores intermediate parses for constituent phrases. The procedure of parsing can be enunciated in the following steps:

1. If n is the number of tokens in a sentence, a chart of size n x n is prepared. In this chart, the position (i,j) would, at the end of the execution of the algorithm, contain all possible parses of the phrase formed by the words from i to j , both inclusive.

2. The entire procedure of parsing consists of many iterations. In each iteration, every entry of all the tokens in the sentence is taken. Its 'sub-category' object is examined to figure out if appropriate neighbouring phrases in the chart can saturate it. In case it can, a new 'Phrase' object is generated and inserted at the appropriate location in the chart.

3. In every iteration, the above process is repeated. Before inserting a 'Phrase', the presence of previous instances of the same phrase is checked for. In case a previous instance exists, there is no sense in re-inserting the 'Phrase' object. Equality of phrases can easily be checked in any programming language with the concept of 'references'.

4. In a particular iteration, if no new phrases have been inserted, the program terminates at the end of that iteration.

5. It can be easily proved that this procedure will terminate. At the end of program execution, the cells in the chart contain complete phrases. Possible structures for the complete sentence can be found at location (n,n).

4.4 Post-Parsing Semantics

Subsequent to the parsing, we invoke semantic considerations to single out the most appropriate parse. Invariably, natural language sentences have more than one parse. It is essential for any further processing that the most appropriate parse be chosen from the multiple parses on the basis of the information present at the sentential level. Again, the parsed output does not have words tagged with the sense in which they are used. It is therefore necessary to apply a full disambiguation procedure to the output.

All of the disambiguation is done by a Word Sense Disambiguator ⟦Sharma *et al.*2002⟧. Different possible senses of a particular word are picked up from WordNet. The parsed output which is sent to this module has the subject pointer for its verbs set to the appropriate noun phrase. The verb can be present as the head of a relative clause or an infinitival phrase or a host of other clauses. The subject field specification is maintained in the lexicon and is language dependent. The subject field is crucial to the operation of the Word Sense Disambiguator module. This module maintains a list of constraints for the sub-categories of the nouns and verbs and subjects of verbs in terms of the ontological categories they belong to. The ontological tree used here has been adapted from

WordNet. It has been marginally augmented to make it applicable to our domain. The system is directly interfaced with WordNet itself, so that in future if additions are made to WordNet, they will be reflected in the operation of our system as well.

The constraints for the sub-categories of the nouns and verbs are not discrete, but fuzzy. The disambiguator module examines all possible combinations of all the senses in each parse, and calculates the credibilities for these sense combinations based on certain statistical and heuristic rules. Credibility values for all constraints are mentioned in the lexicon which forms the basis for the ordering of appropriate parses. Thus a list of parses is obtained in which the nouns and verbs are tagged with their senses. This list is sorted in the order of the credibilities corresponding to the parses. From this list, the most appropriate parse is obtained. The greater the credibility, the more appropriate the particular parse is.

Once the sense disambiguation has been performed it would be relatively simpler to convert these parses into a meaning representation for our larger goal of a unified semantics for English and Hindi texts. These meaning representations are expected to be along the lines of Minimal Recursion Semantics ⟦Copestake *et al.*1995⟧ or Discourse Representation Theory or others, where representations are related to Predicate Logic

5 Results

5.1 Linguistic Analysis

The parser has been tested on a wide range of inputs both in English and in Hindi. As per the lexicons constructed for each of these languages, the parser was successful in obtaining all possible phrase structures for the input sentence. For example, for the sentence `The teacher told the child that he liked that he came`, we successfully obtained both the parses for this sentence, one in which `that` acts like a complementizer and another in which it acts like a relative pronoun.

Similarly for Hindi, the parser has been tested on sentences with a varying degree of complexity. For example, for the sentence राम आम खाता है, the system generates both the possible parses successfully. With the help of the lexicon, it identifies है as the head of the sentence with खाता in its sub-category. खाता has to have two noun phrases as the कर्ता and कर्म arguments. Because of the free word order feature of Hindi, the system assigns both these roles to राम and आम serially. So in the first output, it generates a parse with राम as the कर्ता and आम as the कर्मा, and in the second output, राम as the कर्मा and आम as the कर्ता.

Switching to a more complex example, let's consider राम खाना खाते हुए श्याम को देखता है. This is an ambiguous sentence where the participial adjunct is attached to the phrase श्याम in one reading, and to the phrase राम in the other. The parser generates two outputs as expected.

Let's take one example of a complex sentence in Hindi. Consider, सीता जो गाँव में रहती थी किताबें बेचने के लिये कानपुर गयी The parser finds that the relative clause is modifying सीता and attaches the clause to सीता. The rest of the phrases are constructed as per the grammar mentioned in the lexicon.

5.2 Computational Analysis

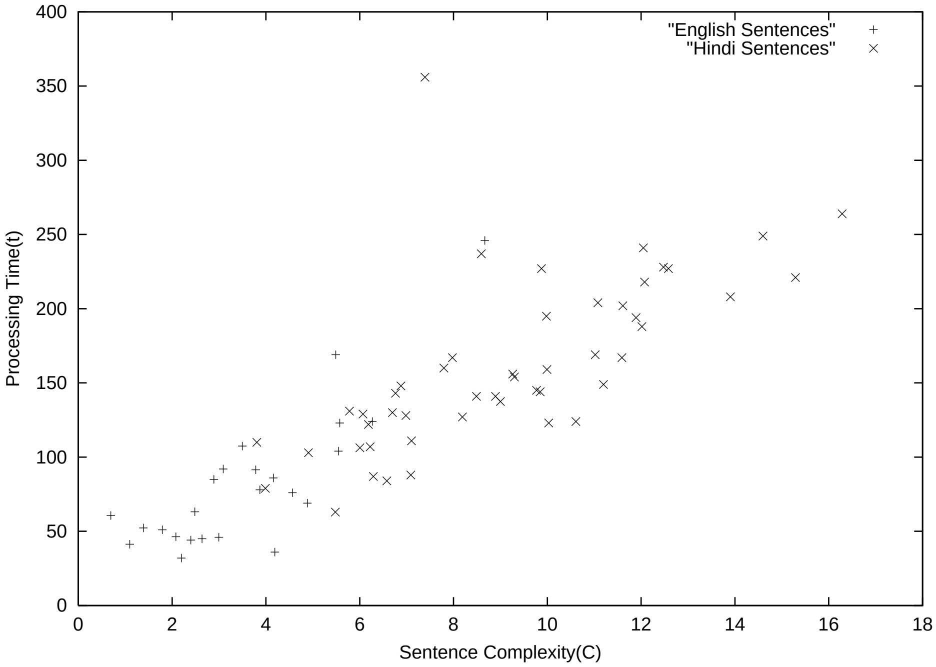

Figure 1: Graph of Processing Time vs Sentence Complexity

With the idea of evaluating the *computational efficiency* of our parser and also to come up with procedures that form a basis for evaluating parsers in general, the concept of *sentence complexity* has been proposed. We use a rough measure of *sentence complexity* which is the natural logarithm of the product of the number of lexical entries for every token in a sentence. Therefore if the sentence can be tokenized as '$t_1\ t_2\ t_3\ \cdots\ t_k$' , and the number of entries of the token t_i in the lexicon is n_i, the proposed *sentence complexity* C is such that

$$C = ln(n_1 n_2 n_3 \cdots n_k)$$

On plotting processing time(t) against C we get a plot as shown in Figure 1. Intuitively one could even use just the product $P = n_1 n_2 n_3 \cdots n_k$ as a measure of sentence complexity, as it in a way signifies the size of the search space in which the parser has to search for legal parses. It might be expected that in the worst case, t is proportional to P. Interestingly, it turns out that our system performs an order of magnitude better than this. The plot of t *vs* $C = ln(P)$ itself comes out to be roughly linear as can be seen in the plot in Figure 1. The same graph shows the nature of the plot for input from both the English and the Hindi corpora that we had created.

5.3 Semantic Analysis

In the semantic module, the task of ordering the obtained parses in order of their 'appropriateness' has been carried out. With the word level constraints, it seems possible to disambiguate the senses and parses of a large number of sentences.

For example, for the sentence `He saw the girl with long hair`, although two parses were generated by the parser, only one of them is appropriate. For the parse in which the phrase `with long hair` is attached to `saw` would mean that `hair` acts as an instrument, which is highly inappropriate. The semantic sense disambiguator recognizes this and assigns a low credibility to this parse. The other parse, in which the prepositional phrase is attached to the `girl`, is given a much higher credibility and is, therefore, selected as the appropriate parse. The corresponding word sense disambiguation for Hindi parsed output has not yet been implemented, due to the unavailability of tools like WordNet and its ontology for Hindi at the moment. The system can however be extended to Hindi as and when such tools become available. The word sense disambiguation lexicon for Hindi can be built along the lines of the corresponding English lexicon.

The credibilities that are in use currently are estimated values. These could be fine tuned to produce better results. As of now we do not have any tools which would enable supervised learning of these values from a large number of inputs, but eventually this would be done using statistical learning tools. However for semantic properties the lexicon needs to be an order of magnitude larger than that used for syntactic purposes. This can be alleviated by using heuristics, eg. assuming that these credibilities are proportional to the instances of the usage of that particular sense combination, as encountered in the corpora. The more number of times a particular sense combination occurs, the more appropriate that combination is. Of course, appropriate scaling factors have to be applied to ensure that these numbers are comparable.

Apart from determining appropriate ordering of parses, we also obtain the sense tags on the words. The ordering of parses obtained can be very neatly exploited in discourse comprehension, where a particular parse might need to be discarded due to inconsistencies in the discourse. In such a case there is always a next appropriate parse which can be considered.

6 Summary and Conclusion

The generalized bilingual parser for English and Hindi yields an output consisting of all possible parses for a given string from either of these languages. The parser is computationally efficient, the processing time being proportional to the *sentence complexity*(C). If the parser can be successfully implemented on a third language, then it can be claimed to be a grammar development environment for a given set of languages. The parser developed here has a semantic module, the word sense disambiguation device, which orders the parsed outputs in a decreasing order of appropriateness. The word sense disambiguator draws on the English WordNet and its ontological classes. A corresponding device for Hindi is expected to be implemented as and when the Hindi WordNet becomes available. The generalized parser developed here is expected to provide input to a unified semantics model for processing of bilingual texts in story-telling domains.

References

[[**Amitabha Mukerjee *et al.*2000**]] Kshitij Gupta Amitabha Mukerjee, Siddharth Nautiyal, Mukesh P. Singh, and Neelkanth Mishra. Conceptual description of visual scenes from linguistic models. http://www.cse.iitk.ac.in/ amit/research/theat.html, 2000.

[[**Carpenter and Penn1994**]] Bob Carpenter and Gerald Penn. Ale the attribute logic engine, 1994.

[[**Chaitanya *et al.*1997**]] Vineet Chaitanya, Amba P. Kulkarni, and Rajeev Sangal. Anusaaraka: Machine translation in stages. *Vivek*, 10, 1997.

[[**Copestake and Flickinger2000**]] Ann Copestake and Dan Flickinger. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Athens, Greece, 2000. citeseer.nj.nec.com/copestake00open.html.

[[**Copestake *et al.*1995**]] Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. Translation using Minimal Recursion Semantics. In *Proceedings of the 6th. International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, Leuven, Belgium, July 1995. citeseer.nj.nec.com/copestake95translation.html.

[[**Erbach1991**]] Gregor Erbach. A flexible parser for a linguistic development environment. In O. Herzog and C.-R. Rollinger, editors, *Natural Language Understanding in LILOG*. Springer, Berlin, Germany, 1991. citeseer.nj.nec.com/erbach91flexible.html.

[[**Fellbaum1998**]] C. Fellbaum. Wordnet:an electronic lexical database, 1998.

[[**Mukerjee *et al.*1999**]] Amitabha Mukerjee, Mahim Mishra, and Rajat Bhattacharjee. A natural language interface for generating video animation. Technical report, Indian Institute of Technology Kanpur, 1999.

[[**Neumann and Flickinger1999**]] G. Neumann and D. Flickinger. Learning stochastic lexicalized tree grammars from hpsg, 1999. citeseer.nj.nec.com/neumann99learning.html.

[[**Pollard and Sag1994**]] Carl Pollard and Ivan E. Sag. *Head-Driven Phrase Structure Grammar*. The University of Chicago press, Chicago, 1994. http://hpsg.stanford.edu/hpsg/hpsg.html.

[[**Sharma *et al.*2002**]] Deepak Sharma, K Vikram, Manav Ratan Mital, Amitabha Mukerjee, and Achla M Raina. Word-sense disambiguation using head-subcat relationship and ontological classes. Suggests an innovative way of word sense disambiguation, 2002.

[[**XTAG1995**]] Research Group XTAG. A lexicalized tree adjoining grammar for english, 1995. citeseer.nj.nec.com/xtag95lexicalized.html.