सार्थक
A bilingual parser for Hindi, English and code-switching structures

P Goyal, Manav R Mital, A Mukerjee,
Achla M Raina, D Sharma, P Shukla, and K Vikram
Indian Institute of Technology, Kanpur,
Kanpur 208016,
Uttar Pradesh, India.
{*pankajgo,manavrm,amit,achla,deepaks,praj,vikram*}@*iitk.ac.in*

Abstract

We present a bilingual syntactic parser that operates on input strings from Hindi and English, as well as code-switching strings drawing upon the two languages. Using the HPSG formalism, we develop grammars for Hindi and English, as well as for the Hindi-English Code-Switching variety (HECS), resulting from contact between these languages in the Indian context. The code-mixed variety under consideration is spoken by Hindi-English ambilinguals in northern India and is regarded as a prestige dialect by the educated elite. Words and larger phrasal constituents from the embedded language are used with the syntax of the matrix language, which is predominantly Hindi. The parser developed here captures this in a lexicon that mixes pure English, pure Hindi, and cross-referenced lexical structures. For ambiguous input, the system generates the set of valid parses, and orders them according to credibility using the ontology derived from WordNet. The system is part of सार्थक, a larger effort aimed at developing a unified semantics for restricted-domain Hindi and English discourse.

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. Generalized parsers that work on more than one language are relatively rare, and mostly work on structurally related languages such as English and French, or English and German ⟦Copestake and Flickinger2000⟧. This paper builds on सार्थक (Saarthaka) ⟦Sharma et al.2002⟧, a bilingual parser for Hindi and English, two structurally disparate languages used in the Indian multilingual context. Here we report work that extends this parser to handle a popular code-mixed variety spoken widely among northern Indian Hindi-English ambilinguals. We shall refer to this variety as Hindi-English Code-Switching (HECS).

The effort is a step towards a natural language understanding system that would operate on bilingual/multilingual texts set in a limited domain story context. The larger aim is to provide multilingual tools for animating textual stories in Hindi and English ⟦Mukerjee et al.2000⟧. The philosophical motivation lies in the relationship posited between understanding of textual input and mental imagery, which is here captured using graphic animation.

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⟧, an HPSG grammar of Hindi has been developed here to provide the basis for the generalized parser. This grammar has been extended to handle structures of the code-mixed variety under consideration, HECS. The bilingual parsing system works on an invariant set of principles that encompass Hindi, English, and HECS grammars. The code-switching structures are handled through cross-referencing and additional subcategories in a merged lexicon.

An important feature of सार्थक is that it provides not only all possible parses for each string, but also an ordering between them. This is done by a word sense disambiguator, which uses the ontology derived from WordNet ⟦Fellbaum1998⟧ to arrange the output in a decreasing order of appropriateness.

In its current phase of implementation, the parser operates on pure English and pure Hindi input strings, as well as code-switching strings from HECS.

1.1 Hindi-English Code-Switching (HECS)

In India where English is a prestige language, code-mixed varieties drawing upon English as the embedded language assume a degree of prestige:

> ". . . the mixed language can be said to have prestige, since the amount of mixing corresponds with the level of education and is an indicator of membership in the elite group – Annamalai ⟦Annamalai2001⟧.

HECS is part of the speech repertoire of Hindi-English ambilinguals, who switch codes in different speech contexts for sociolinguistic reasons (e.g. to signal social distance/proximity in interactions) or for purely linguistic reasons (e.g. to fill a lexical gap in the matrix language). HECS is a stable variety which demonstrates a certain regularity in observing constraints on structure. Adopting the terminology from Peter Auer, it may be referred to as a "Fused Lect" as opposed to a "Mixed Code" ⟦Auer1998⟧.

Let us consider some examples from HECS:

(1) student ने teacher के लिये library से
book issue की .
student CM teacher CM library CM book issue
operator-tense.
The student issued a book from the library
for the teacher.
(2) raama whom I saw yesterday आज दोबारा
मिला.
raama RM I saw yesterday today I
CM again meet-tense.
Today, I again met Ram, whom I had
seen yesterday.

The sentence (1) is constituted almost entirely of English lexical items, yet the usage is perfectly normal in most North Indian contexts. The verb "issue" is drawn from English to fill a lexical gap in Hindi. The code-switching verb "issue की(kii)", consisting of the English verb "issue" and a form of the Hindi operator "करा" (kara), observes the Hindi grammatical restrictions: each of the noun phrases occuring in the string is followed by postpositions "ने(ne)","के लिये(ke liye)", and "से(se)" - as required by Hindi grammar. These Hindi grammatical restrictions do not apply when the English verb "issue" is used as the verbal head in a code-switching context. Thus strings such as

(3)*student ने issued a book teacher
के लिये library से.
(4)*छाञ issued किताब एक अध्यापक
के लिये पुस्तकालय से.

which use English verbs are unacceptable in HECS. Sentence (2) consists of a Hindi head-verb with two noun phrases, of which one, "राम(raama)", contains an English relative clause in it. This code-switching string appears to observe restrictions imposed by Hindi grammar. It may be noted that whereas in the corresponding English version, a tense sequencing constraint on the verb requires the form had seen in the relative adjunct, the code-switching string does not observe this constraint.

1.2 Parsing Code-Switching Structures

The study of code-mixing has gained immense popularity in recent years. Although much work has been done on grammatical analyses of different code-mixed varieties used in South Asia ⟦Joshi1985, Annamalai2001⟧, and around the world ⟦MacSwan1997, Mahootian and Santorini1995⟧, implemented grammatical analyses of code-mixed varieties are scarce.

Constraints on code-switching have been a subject of discussion ever since the earliest proposals regarding grammatical properties of code-mixed varieties began to appear in the 70's. Most researchers agree that code-mixing is governed by a "third grammar" which constrains the interaction of the two language systems. The important question to investigate is whether machine implementation would require explicit construction of a completely separate "third" grammar (and its associated lexicon), or whether this can be achieved by suitable extensions to the existing grammars of the matrix and embedded languages.

Some of the earlier constraints cited in the literature (following ⟦MacSwan1997⟧) include

- Free Morpheme Constraint: There may be no switch between a bound morpheme and a lexical form unless the lexical form has been phonologically integrated into the language of the bound morpheme ⟦Sankoff and Poplack1981⟧.
- Equivalence Constraint: Code switching can occur at points in discourse when the juxtaposition of L1 and L2 elements does not violate a syntactic rule of either language ⟦Poplack1980⟧.
- Dual Structure Constraint: Structure of the embedded constituent need not conform to the constituent structure rules of the matrix language, so long as the placement in the matrix language obeys the rules of the matrix language ⟦Sridhar and Sridhar1980⟧.

Joshi (1985) proposes the Closed-Class constraint which stipulates that closed class items (det., Q-word, preposition, auxiliaries etc.) from one language cannot be mixed with open class items from the other. Mahootian and Santorini (1995) propose alternate accounts that focus on the Head-complement relation in the sentence. In our work, we adopt an approach similar to Mahootian and Santorini, and and constrain the grammar such that the lexical requirements of heads determine the structure in code-switching strings. In other words, phrasal heads determine the syntactic properties of the subcategorized elements, regardless of the language from which these elements are drawn. This position is also close to the Word Grammar Integrity Corollary proposed by Belazi et al ⟦Belazi et al.1994⟧ which stipulates that "A word of language X, with grammar GX, must obey grammar GX".

Researchers who challenge a constraint-based approach to code-mixing also question attempts to assign different constituents of a code-mixed variety to the so called matrix and embedded languages, suggesting that such assignment is largely a matter of perspective ⟦Agnihotri1998⟧. Agnihotri cites data that show violations of the constraints proposed in the literature. He goes on to suggest that "the concept of 'a language' with its attendant codified grammar may not be adequate for characterizing the constraints that condition the nature of mixed codes . . ." ⟦Agnihotri1998⟧(p. 228).

Our work Hindi-English code mixing attempts to capture some of the fuzziness inherent in code-mixed varieties within the broader objective of developing a machine implementable parser for the mixed code, HECS. This is done through a merged lexicon with cross-linkings between English-Hindi Synsets. Also, we adopt a very general structural constraint which invokes the subcategorization properties of the lexical heads, regardless of the source language. As such, in our system notions like matrix and embedded language are simply a matter of terminological convenience.

2 Parser Implementation

Most parsers in use today ⟦Copestake and Flickinger2000, Carpenter and Penn1994⟧ are not general enough to incorporate more than one language. However, systems such as LKB ⟦Copestake and Flickinger2000⟧ and LilFes or LILOG ⟦Erbach1991⟧ or ⟦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.

Saarthaka implements a Hindi-English bilingual parser that draws upon two separate lexicons of Hindi and English. In handling pure Hindi or pure English input strings, one or the other lexicon is chosen based on the words of the input string. The question that arises while handling HECS is whether to introduce a third mixed-code lexicon, or whether to add appropriate structures to an union of the existing lexicons. The merged-lexicon approach appears to be more elegant than a third grammar option, and also addresses the fact that speakers of HECS are fluent in both Hindi and English and are actually drawing upon both the grammars to create a new fused lect.

Existing parsers based on commonly employed grammatical formalisms like the ATN, TAG, CFG, and HPSG rarely deal with multilingual syntactic and semantic issues. सार्थक is an advance to the extent that it simultaneously operates on a bilingual input, handles code-switching structures, and, that it generates a list of parses in decreasing order of appropriateness. Parsers such as LKB incorporate Minimal Recursion Semantics ⟦Copestake et al.1995⟧, but our treatment is directly linked to the WordNet ⟦Fellbaum1998⟧, which provides access to a large body of semantic data.

Another innovation is that unlike other implementations of HPSG which are built upon logic languages such as Prolog, our parsing engine is designed for portability and is built grounds-up in Java.

3 Methodology

3.1 Domain

Saarthaka's larger aim, that of generating graphics animation from multilingual stories, requires the use of a restricted domain within which object graphic models and verb action procedures can be instantiated. At present, the work is restricted to stories involving a typical family environment, with about 400 words in English and a little less in Hindi. The Devanag transliteration scheme has been used for representing the Hindi text.

In the domain under consideration, we found it sufficient to use only a subset of the HPSG feature sorts. However, some features needed to be extended to incorporate aspects such as obliqueness in Hindi grammar. For example, the noun कुत्ता can be used either in its root form or in the oblique form, which is captured by a new feature in the fifth subfield in its entry:

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

In the same spirit, verbal agreement for gender is essential in Hindi, and is implemented by inserting a new field in verb entries.

3.2 Grammar Generation

Since the HPSG grammar remains relatively unexplored for South Asian languages, we have developed our own grammar for Hindi within the broad HPSG framework. Even for English, owing to the larger objectives that require a semantic treatment, we have constructed a grammar and lexicon of our own. The needs of the code-mixed variety HECS have led to further modification of the lexical and grammatical structures.

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. In order for the same parsing engine to handle Hindi and English inputs, the design of the lexicon needs to specify constrained word-order grammars (as in English) as well as free word-order grammars (as in Hindi).

As an example, consider the following entry in the non code-mixed English lexicon,

`dog:dog` N(-,S,-,C) { D(S) | *J | ! }

The word following the colon is the root word, used in semantic processing (here it is the same as `dog`). This is followed by the lexical category, with a list of the case, number, gender, property. The second list is the set of features which can exist in the sub-category of this head word - here it 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 | ! }

Here the notation '∼' indicates that the determiner is not compulsory. Again, the entry for `eats` would be

`eats:eat` V(P,S) { N(N,S,-,-) | ! | N(A,-,-,-) }

Let us now consider the entry ghara (house) from the Hindi lexicon:

घर N(N,3,S,M,-,V) { ∼D(S,M) | ∼J(S,M)
| ∼k(S,M) | ! | }

The description of Hindi nouns consists of six features as compared to four in English; the additional features in Hindi are Obliqueness and the Person.

For the Hindi verb entry khaata, consider:
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 delimiters '/' and '\' indicate that all features occuring inside them are freely ordered in the structure of which this word is the head. The notation '[' and ']' indicate complimentarity - i.e. only one of the features inside them can occur in the sub-category. Note that the subcategorization also differs on a few parameters which are unique to each language.

The lexicon has entries for expressions that might consist of more than one word. For instance the expression '`in front of`' in English and 'के लिये' in Hindi have independent entries.

3.3 HECS Grammar Generation

Multilingual parsers permit text in one of several languages to be input, and discriminate between them at run-time. Such systems typically employ different lexicons for each language, which are identified based on the lexical items used.

Mixed-code parsers could be built using separate lexicons for each of the input languages, or by merging the lexicons. The merged lexicon option has an obvious advantage over separate lexicons in terms of economy and elegance. Among others, Macswan ⟦MacSwan1997⟧ has claimed a merged lexicon as a more viable option for mixed-code grammars.

However, implementing a grammar based on a merged lexicon has the problem that the syntactic constraints of one language are applied to lexical items from the other, thus generating incorrect parses. Consider the following example:

(5) raama who lives in the house books
बेचने के लिये कानपुर गया
becane ke liye kaanpura gayaa
to sell books went to Kanpur

Here the parser based on a merged lexicon, permits the word "books" to accept "in the house" as a subcategory based on its english lexical structure. This results in two parses, including the following incorrect one:

```
       +-raama who lives in the house
books becane ke liye kaanapura
gayaa
       +-raama who lives
       | +-raama N(N,S,M,-)
       | +-who lives
       |   +-who W(-)
       |   +-lives H(P,S,V,N)
       +-in the house books becane
ke liye
       | +-in the house books becane
       | | +-in the house books
       | | | +-in the house
       | | | | +-in P(N)
       | | | | +-the house
       | | | |   +-the_D(-,-)
       | | | |   +-house_N(-,S,-,-)
       | | | +-books_N(-,P,-,-)
       | | +-becane_Z(P,-)
       | +-ke liye_p()
       +-kaanapura_N(A,3,S,M,-,V)
       +-gayaa_A(-,-,S,M,-,V)
```

English being a verb medial language, the first noun phrase - in this case "raama who lives in the house"- should ordinarily be followed by a verb phrase, introduced by a verbal rather than a nominal element. But in the code-switching string above, it is the Hindi purposive clause "books becane ke liye"- beginning with a nominal element "books" - that follows the first noun phrase. Now since "books" has features specified by the English grammar, the parser sees it as a nominal head subcategorizing the prepositional phrase "in the house", as in a sring like "in my house books are kept in the study", thus generating the incorrect parse above.

One obvious solution to this problem would be to abandon the merged lexicon option, and create a third grammar and associated lexicon for the code-mixed lect. The other solution is to retain a merged lexicon and introduce an additional feature field indicating language - 'H' or 'E'. To handle the above, 'books' is cross-linked with किताबें (kitaabe.m), and takes an 'H' feature, and it does not unify with the preceding phrase "in the house". The original entry of "books", marked 'E' is not operative since the head verb is Hindi. Thus only one possible parse is generated:

```
       +-raama who lives in the house
books becane ke liye kaanapura
```

```
gayaa
        +-raama who lives in the house
        | +-raama_N(N,3,S,M,-,V,H)
        | +-who lives in the house
        |   +-who_W(-)
        |   +-lives in the house
        |     +-lives_H(P,S,V,N)
        |     +-in the house
        |       +-in_P(N)
        |       +-the house
        |         +-the_D(-,-)
        |         +-house_N(-,S,-,-,E)
        +-books becane ke liye
        | +-books becane
        | | +-books_N(A,3,P,F,1,V,H)
        | | +-becane_Z(P,-)
        | +-ke liye_p()
        +-kaanapura_N(A,3,S,M,-,V,H)
        +-gayaa_A(-,-,S,M,-,V)
```

Based on these considerations, in our work we have used a merged lexicon with cross-linking between English-Hindi synsets, as well as source language tags in the cross-linked entries.

3.4 HECS Parsing

The structure in mixed code is governed by the head verb in the sentence, and correspondingly, by other heads in the rest of the phrases. Thus, if the head is from Hindi, the structure observes constraints imposed by the Hindi grammar (and similarly for English). Given this broad constraint on code-mixing, three specific mechanisms are used to handle code-switching in HECS.

1. **Noun substitution**
   Noun substitutions are handled as equivalence relations between Hindi and English words that function as synsets in HECS. Thus an entry for the word ’books’ is equated to the list of fields for किताबें, किताबों which ’books’ can now replace in code-switching sentences such as:
   [उसने books/किताबों को पढ़ा]
   or
   [raama books/किताबें पढ़ता है]

   The two additional entries for books are:
   किताबें,books :kitaab,book N(A,3,P,F,1,V,H)
   किताबों,books :kitaab,book N(N,3,P,F,0,V,H)

   We note also that the entry for “books” with feature ’H’ inherits the PNG and oblique features of the hindi noun “किताबों” which then constrain its usage.

   Similarly, the Hindi word किताबें is equated with the entry of the English word ’books’ as
   books,kitaabe.m:book,kitaab N(-,P,-,-,E).
   This enables us to parse sentences such as
   [I have issued these kitaabe.m]

2. **Replacement of Hindi verbs by English**
   The Hindi verb can be replaced by an English root verb plus the Hindi operator कर (kara), and its morphological variants such as किया,की,करता (’kiyaa’,’kii’,’karataa’) etc. For example - the sentence
   [उसने किताब पढ़ी]
   can be replaced by
   [उसने किताब read की]
   In such cases the head verb, instead of पढ़ी, remains the Hindi की, which takes on an additional subcategory (the English root verb ”read” in the example). So the original entry for की (kii):
   की : की A(-,-,S,M,-,V) {/n(V) | [N(A,-,S,M,1,V)| a(V)] | ∼ i |∼ p |∼ s |∼ l |∼ Y \ | !}
   now changes into की:की A(-,-,S,M,-,V){/n(V)|[N(A,-,S,M,1,V,H)| a(V)]| ∼i| ∼ p | ∼s | ∼ l | ∼ Y|∼ RV\ |!}
   where English root verb entries are tagged as “RV” so that the operator ’kara’ can now take them as subcategories:
   `read:read RV!`

3. **Mixed Phrasal Constituents**
   Mixed phrasal constituents or clausal adjuncts depend on a *hinge word* which in the main structure is being used as the matrix language word, and in the phrasal constituent is being used as an embedded language word.

   Consider the sentence (5) again:
   (5) raama who lives in the house books बेचने के लिये कानपुर गया
   Here, “raama” as the head of a subcategory of गया (gayaa) is a Hindi word, yet in the phrasal constituent “raama who lives in the house” it appears as English - and thus serves

as the hingeword.

Similarly, in relative clauses, relative markers such as Hindi जो (jo) or English “which”, “who” etc. act as the hingewords. In clausal adjuncts, markers such as English “having” or Hindi “-कर” serve the hinge function.

Thus in order to parse mixed phrasal constituents we need to introduce multiple entries for the hingewords tagged as ’H’ or ’E’. For example in sentence (5), “raama”, as a Hindi Noun has to accept a subcategory with ’lives’ as its head. This is done using the entry:
राम:राम N(N,3,S,M,-,V,H){∼S(-,-,-,M,-,-)| ∼k(-,M)|*J(S,M)|!|[∼H(-,S) | ∼Q(N,-,S,M,1,W)| ∼P(N,-,S,M,1,W)| ∼A(N,-,S,M,-,W)| ∼Q(A,-,-,-,1,W)| ∼P(A,-,-,-,1,W)| ∼A(A,-,-,-,-,W)]}
here the subcategory ∼H(-,S) (where H stand for the verbs which act as heads to relative phrases) has been added which then takes ’who’ and ’in the house’ as its English subcategories. Now consider the predominantly English form of (5) -
(5b) raama जो घर में रहता हैwent to Kanpur to sell books.
raama jo ghara me.m rahataa hai
raama who lives in the house
where raama being a subcategory of ’went’ is of type ’English’ or ’E’ and does not take hindi subcategories. To handle this, the reconstructed subcategory of ”raama” looks like -
raama:raama N(-,S,M,-,E){[∼C | ∼P(N)| ∼G | ∼K | ∼Y]|*J|!| ∼H(-,S)| ∼Q(N,-,S,M,1,W)}
Here the subcategory ∼Q(N,-,S,M,1,W)(where Q stands for the Hindi auxiliary है) has been added which then takes ’rahataa’ in its subcategory.

3.5 Parser Implementation

Parsing involves searching for all possible phrase combinations which can unify with the grammar to yield a sentence. In HPSG terms, it amounts to finding the heads of all constituent phrases in a given sentence and the phrases that saturate the subcategories of these heads. With sentences that are ambiguous all possible parses should be generated.

We apply a ‘Chart Parsi ng’ approach in which a chart is maintained which stores intermediate parses and avoids re-visits. The procedure consists of 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) at the end of the execution of the algorithm, contains all possible parses of the substring from word i to word j.

2. The parsing process iterates over every entry for all the tokens in the sentence. Its ‘subcategory’ object is examined to figure out if appropriate phrases in the chart can saturate it. If it succeeds, and if the generated ‘Phrase’ is not present in the Chart, it is inserted at the appropriate location in the chart. Equality of phrases are checked with the concept of ‘references’.

3. The program terminates if no new phrases have been inserted at the end of a particular iteration.

4. It can be easily proved that this procedure will terminate. At the end, possible structures for the complete sentence are found at location (n,n).

4 Summary and Conclusion

The Hindi-English bilingual parser developed here yields an output consisting of all possible parses for strings from either of these languages and from the code-mixed variety HECS. Subsequent to the parsing, सार्थक (Saarthaka) orders the parses in terms of their appropriateness through a Word Sense Disambiguator ⟦Vikram et al.2002⟧ using the ontological classes derived from WordNet. This ordering of possible parses has not yet been implemented for Hindi and the mixed code HECS, due to the unavailability of tools like WordNet for Hindi at the moment. The system can however be extended to structures from Hindi and HECS as and when such tools become available.

The work assumes a general constraint on code-mixing which has been stated as follows: Phrasal heads determine the syntactic properties of the subcategorized elements, regardless of the language from which these elements are drawn. This constraint is implemented through a lexicon that contains entries from the matrix and embedded languages, as well as cross referenced items from both. Parsing of code-switching structures is achieved by building constraints of the Hindi phrasal heads into the lexicon such that they apply not only to Hindi subcategorized constituents but also to those drawn from English.Similar modifications are done for the case when English is the matrix language.

References

R.K. Agnihotri. 1998. *Social Psychological Perspectives on Second Language Learning*. Sage Publications.

E. Annamalai. 2001. *Managing multilingualism in India – Political and Linguistic manifestations.* Sage Publications, New Delhi.

Peter Auer. 1998. From code-switching via language mixing to fused lects: Toward a dynamic typology of bilingual speech. Technical Report InLiSt No. 6, Interaction and Linguistic Structures, Freiburg i. Br., September.

Hedi M. Belazi, Edward J. Rubin, and Almeida J. Toribio. 1994. Code switching and x-bar theory: The functional head constraint. *Linguistic Inquiry*, 25(2):221–237.

Bob Carpenter and Gerald Penn. 1994. Ale the attribute logic engine.

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

Ann Copestake and Dan Flickinger. 2000. 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.

Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. 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.

Gregor Erbach. 1991. A flexible parser for a linguistic development environment. In O. Herzog and C.-R. Rollinger, editors, *Natural Language Understanding in LILOG*. Springer, Berlin, Germany.

C. Fellbaum. 1998. Wordnet:an electronic lexical database.

Aravind Joshi. 1985. Processing of sentences with intrasential code switching. In *Natural Language Parsing: Psychological, Computational and Theoretical Perspectives*. Cambridge University Press, Cambridge.

Jeffrey MacSwan. 1997. *A Minimalist Approach to Intrasentential Code Switchi Spanish-Nahuatl Bilingualism in Central Mexico*. Ph.D. thesis, University of California Los Angeles.

Shahrzad Mahootian and Beatrice Santorini. 1995. Codeswitching and the syntactic status of adnominal adjectives. *Lingua*, 95:1–27.

Amitabha Mukerjee, Kshitij Gupta, Siddharth Nautiyal, Mukesh P. Singh, and Neelkanth Mishra. 2000. Conceptual description of visual scenes from linguistic models. *Journal of Image and Vision Computing, Special Issue on Conceptual Descriptions*, 18(2):173–187.

Carl Pollard and Ivan E. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

S. Poplack. 1980. Sometimes i start a sentence in english y termino en espanol: Towards a typology of code- switching. *Linguistics*, 18:581–618.

D. Sankoff and S. Poplack. 1981. A formal grammar for code-switching. *Papers in Linguistics*, 14(1):3–46.

Deepak Sharma, K. Vikram, Manav R. Mital, Amitabha Mukerjee, and Achla M Raina. 2002. Saarthaka - an integrated discourse semantic model for bilingual corpora. In *Proc. Intl Conf on Universal Knowledge and Language*, Goa India, Nov.

K.N. Sridhar and S.N. Sridhar. 1980. Psycholinguistics of bilingual code-mixing. *Canadian Journal of Pyschology*, 34(4):409–418.

K Vikram, Deepak Sharma, Manav R. Mital, Amitabha Mukerjee, and Achla M Raina. 2002. An hpsg and wordnet based approach to word sense disambiguation. In Kavitha M Sasikumar M, Jayaprasad Hegde, editor, *Artificial Intelligence: Theory and Practice, KBCS-2002*, Mumbai, India, December. Vikas Publishing House.