Machine Translation **10:** 293–319, 1995.

Parsing Turkish Using the Lexical Functional Grammar Formalism

ZELAL GÜNGÖRDÜ*
Centre for Cognitive Science, University of Edinburgh, 2 Buccleuch Place, Edinburgh, EH8 9LW Scotland, U.K., gungordu@cogsci.ed.ac.uk

and

KEMAL OFLAZER
Department of Computer Engineering and Information Science, Bilkent University, Ankara 06533, Turkey, ko@cs.bilkent.edu.tr

Abstract. This paper describes our work on parsing Turkish using the *lexical-functional grammar* formalism [11]. This work represents the first effort for wide-coverage syntactic parsing of Turkish. Our implementation is based on Tomita's parser developed at Carnegie Mellon University Center for Machine Translation. The grammar covers a substantial subset of Turkish including structurally simple and complex sentences, and deals with a reasonable amount of word order freeness. The complex agglutinative morphology of Turkish lexical structures is handled using a separate two-level morphological analyzer, which has been incorporated into the syntactic parser. After a discussion of the key relevant issues regarding Turkish grammar, we discuss aspects of our system and present results from our implementation. Our initial results suggest that our system can parse about 82% of the sentences directly and almost all the remaining with very minor pre-editing.

Key words: Parsing, Natural Language Grammar Development, Turkish, Lexical Functional Grammar

1. Introduction

As part of our ongoing work on the development of computational resources for natural language processing in Turkish, we have undertaken the development of a parser for Turkish using the lexical-functional grammar formalism for use in a number of applications. Parsing is possibly the most important component of any natural language processing application, such as a machine translation system or a natural language database interface system. Parsing performs an analysis of the natural language utterance or sentence, resolving ambiguities to the extent possible using various sources of information ranging from simple lexicons, to sophisticated statistical language models, and world models in addition to linguistic information. It extracts structural and semantic information from the input that can then be used in other stages of the application. Although there have been a number of studies of Turkish syntax from a linguistic perspective (e.g., [15]), this work represents

* This work was done as a part of the first author's M.Sc. degree work at the Department of Computer Engineering and Information Science, Bilkent University, Ankara, 06533, Turkey.

the first approach to the wide-coverage computational analysis of Turkish syntax using a well-established contemporary linguistic theory. Our implementation is based on Tomita's parser developed at Carnegie-Mellon University Center for Machine Translation [17, 28]. Our grammar covers a substantial subset of Turkish including structurally simple and complex sentences, and deals with a reasonable amount of word order freeness. This system is expected to be a part of the machine translation system that we are planning to build as a part of a large scale natural language processing project for Turkish, supported by NATO [22].

Turkish has two characteristics that have to be taken into account: agglutinative morphology, and rather free word order with explicit case marking. We handle the complex agglutinative morphology of the Turkish lexical structures using a separate morphological processor based on the two-level paradigm [1, 13, 21] that we have incorporated into the lexical-functional grammar parser. Word order freeness, on the other hand, is dealt with by relaxing the order of phrases in the phrase structure parts of lexical-functional grammar rules by means of generalized phrases, and letting case features of NPs signal their grammatical roles rather than their positions in the phrase structure.

The ATMACA system by Stoop [27] represents an earlier effort for parsing Turkish sentences. It is claimed to be a semantic parser based on the previous work by the same author, based on a context-free grammar with about 13 rules and using a case-frame representation as the semantic representation. No information about the coverage of its lexicon and grammar is provided.

After a brief overview of the lexical-functional grammar formalism in Section 2, in Section 3 we present a summary of the salient features of the Turkish language, especially relevant to natural language processing. We then present the architecture of our parser in Section 4, and discuss a number of important points about the grammar developed, in Section 5. This is followed by a summary of the results of our evaluation of the parser on Turkish text in Section 6, and then a number of examples highlighting the capabilities of the parser in Section 7. Finally, in Section 8, we present our conclusions and discuss a number of ways to improve the performance of the system further.

2. Lexical-Functional Grammar

Lexical-functional grammar (LFG) [11] is a linguistic theory which fits nicely into computational approaches that use *unification* [26]. Because of space limitations, here we present only some of the formal highlights of the theory, summarizing them from Kaplan and Bresnan [11]. One can refer to this work for a complete description of the formal principles of the theory, and to the other chapters in Bresnan [2] for its extensive linguistic and psychological motivation.

In a lexical-functional grammar, the syntactic structure of every sentence of a language is encoded in two parallel levels of syntactic representation: a *constituent structure (c-structure)* and a *functional structure (f-structure)*. Using two different

levels of syntactic representation enables LFG to separate information about the grammatical functions in a sentence from its phrase structure. C-structures represent phrase structure configurations, in the form of a conventional phrase structure tree, defined in terms of syntactic categories, terminal strings and their dominance and precedence relationships. F-structures represent information about the grammatical relations between parts of sentences as sets of pairs of attributes and values. Attributes may be feature names (such as TENSE, NUMBER, CASE and PRED) with values of kind *simple (atomic) symbols* (such as PAST, SING and ACC) or of kind semantic forms (which are indicated as the value of the PRED feature and govern the process of semantic interpretation), or grammatical function names (such as SUBJECT and OBJECT) with values of kind *subsidiary f-structures.*

The c-structure of a string is generated by a context-free c-structure grammar that is augmented using functional specifications (*functional schemata*), which indicate how the functional information contained on a node in the c-structure participates in the f-structure of the left hand side constituent. Lexical entries are also enriched by such functional schemata, which determine their syntactic features and semantic content. Functional schemata (associated with both the phrase structure rules and the lexical entries) provide the information needed to construct the f-structure of a string. (See Kaplan and Bresnan [11] for details.)

There are three well-formedness conditions on f-structures:

1. **Uniqueness Condition:** In a given f-structure, a particular attribute may have at most one value.
2. **Completeness Condition:** An f-structure is *locally complete* if and only if it contains all the governable grammatical functions that its predicate governs.[1] An f-structure is *complete* if and only if it and all its subsidiary f-structures are locally complete.
3. **Coherence Condition:** An f-structure is *locally coherent* if and only if all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is *coherent* if and only if it and all its subsidiary f-structures are locally coherent.

A string is grammatical only if its f-structure satisfies the uniqueness, completeness and coherence conditions.

3. Turkish Grammar

In this section, we highlight two of the relevant key issues in Turkish grammar, namely highly inflected agglutinative morphology and free word order, and give a description of the structural classification of Turkish sentences that we deal with.

[1] For any given language, some function G is a member of the set of *governable grammatical functions* if and only if there is at least one semantic form that subcategorizes for it [25]. A given lexical entry mentions only a few of the governable functions; this lexical entry is said to govern these functions.

3.1. MORPHOLOGY

Turkish is an agglutinative language with word structures formed by productive affixations of derivational and inflectional suffixes to root words [21]. This extensive use of suffixes causes morphological parsing of words to be rather complicated, and results in ambiguous lexical interpretations in most cases. For example:[2]

(1) **çocukları**
çocuk+lar+ı

a. child+PLU+3SG–POSS 'his children'
b. child+PLU+ACC 'children' (accusative)
çocuk+ları
c. child+3PL–POSS 'their child'
d. child+(PLU)+3PL–POSS 'their children'

Such ambiguity can sometimes be resolved at phrase and sentence levels by the help of agreement requirements though this is not always possible. Example (2) explores the help of two agreement requirements in Turkish, in eliminating morphological ambiguity in syntactic level. These are possessor specifier–head noun agreement in possessive noun phrases and subject–verb agreement, which essentially follow the same pattern.[3] In (2a) only the interpretation (1d) above (i.e., *their children*) is possible because:

- possessor specifier–head noun agreement and subject–verb agreement rule out (1a) and (1c), respectively, and
- the facts that the verb *gel- (come)* does not subcategorize for an accusative marked direct object, and that in Turkish the subject of a finite sentence must be nominative (i.e., unmarked) rule out (1b).

(2a) **O+nlar+ın** **çocuk+ları** **gel+di+ler.**
it+PLU+GEN child+(PLU) come+PAST+3PL
(they) +3PL-POSS
'Their children came.'

In (2b), on the other hand, both (1a) (i.e., *his children*) and (1d) (i.e., *their children*) are possible since the possessor of the noun *çocukları*, is a covert one: it may be either *onun* (*his*) or *onların* (*their*). The other two interpretations are ruled out due to the same reasons as in the case of (2a).

[2] Turkish is a pro-drop language. Pronominal possessors and subjects are usually dropped.

[3] The possessor agrees in person and number with the possessive suffix on the head noun; a third person plural possessor can agree with both a third person singular and a third person plural possessive suffix on the head noun. Subject–verb agreement follows the same pattern with the possessive suffix being replaced by the verbal agreement suffix in finite sentences.

(2b)	**Çocukları**	**geldiler.**	
	Çocuk+lar+ı	**gel+di+ler.**	'His children came.'
	child+PLU+3SG–POSS	come+PAST+3PL	
	Çocuk+ları	**gel+di+ler.**	'Their children came.'
	child+(PLU)+3PL–POSS	come+PAST+3PL	

3.2. Word Order

In terms of word order, Turkish can be characterized as a *subject–object–verb (SOV) language* in which constituents at some phrase levels can change order rather freely. This is due to the fact that morphology of Turkish enables morphological markings on the constituents to signal their grammatical roles without relying on their order. This, however, does not mean that word order is immaterial. Sentences with different word orders reflect different pragmatic conditions, in that, topic, focus and background information conveyed by such sentences differ.[4] Besides, word order is fixed at certain phrase levels such as postpositional phrases. There are even severe constraints at sentence level, some of which happen to be useful in eliminating potential ambiguities in the interpretation of sentences.

One such constraint is related to the existence of case marking on direct objects. Direct objects in Turkish can be either accusative marked or unmarked (i.e., nominative). The existence of case marking generally correlates with a specific reading of the object [5].[5] The constraint is that nominative direct objects can only appear in the immediately preverbal position in a sentence, which determines that *mutluluk* is the subject and *huzur* is the direct object in (3):[6]

(3)	**Mutluluk**	**huzur**	**getir+ir.**
	happiness	peace of mind	bring+PRES(+3SG)
	'Happiness brings peace of mind.'		
	*'Peace of mind brings happiness.'		

Another constraint is that non-derived manner adverbs[7] always immediately precede the verb or, if it exists, the nominative direct object [6, pages 192–196]. Hence, *iyi* can only be interpreted as an adjective that modifies the accusative direct object *yemeği* in (4a), whereas in (4b), it is an adverb modifying the verb *pişirdin*.

[4] See Erguvanlı [6] for a discussion of the function of word order in Turkish grammar.

[5] See Nilsson [19] for a more general discussion of the function of case-marking, including accusative marking, in Turkish.

[6] This example is taken from Erguvanlı [6].

[7] The term "non-derived" in this context refers to the fact that these adverbs have not gone through any of the adverb derivation processes in Turkish, such as re-duplication, suffixation (e.g., of the suffixes -ce, -le (and their allomorphs), -leyin, etc.), or a combination of these two processes [6, pages 183–186]. They are in fact qualitative adjectives, but can also be used as adverbs. Examples are *iyi 'good/well'*, *hızlı 'fast'*, *güzel 'beautiful/beautifully'*.

In (4c), on the other hand, it can either be an adjective modifying the nominative direct object *yemek*, or an adverb modifying the verb *pişirdin*:

(4a) **İyi** **yemeğ+i** **pişir+di+n.**
good meal+ACC cook+PAST+2SG
'You cooked the good meal.'
*'You cooked the meal well.'

(4b) **Yemeğ+i** **iyi** **pişir+di+n.**
meal+ACC well cook+PAST+2SG
'You cooked the meal well.'

(4c) **İyi** **yemek** **pişir+di+n.**
good/well meal cook+PAST+2SG
'You cooked a/some good meal.'
'You cooked well.'

There are also particular pragmatic conditions that govern the selection of the most felicitous word order in a given context [6, 7]. We will not go into details of the pragmatic conditions conveyed by different word orders, but will rather provide a number of examples for such conditions. (See Erguvanlı [6] for a thorough discussion of these conditions.) For instance, a constituent that is to be emphasized is generally placed immediately before the verb. For example, (5a) is an example of the typical word order whereas in (5b) the subject, *ben*, is emphasized.[8] In (5c), on the other hand, the indirect object, *çocuğa*, is emphasized:

(5a) **Ben** **çocuğ+a** **kitab+ı** **ver+di+m.**
I child+DAT book+ACC give+PAST+1SG
'I gave the book to the child.'

(5b) **Çocuğ+a** **kitab+ı** **ben** **ver+di+m.**
child+DAT book+ACC I give+PAST+1SG
'I gave the book to the child.'

(5c) **Ben** **kitab+ı** **çocuğ+a** **ver+di+m.**
I book+ACC child+DAT give+PAST+1SG
'I gave the book to the child.'

[8] The underlined words in Turkish examples show the constituent that is emphasized and the ones in English translations show the word marked with stress.

In addition, in some contexts one or more constituents may appear in the post-predicate position. Post-predicate constituents are in general discourse-predictable or recoverable [6, chapter 2]. For example, (6) would be a felicitous utterance in a context where the addressee was a day late for her appointment:

(6) **Dün** **gel+meli+ydi+n** **bura+ya.**
yesterday come+NECC+PAST+2SG here+DAT
'You should have come here yesterday.'

3.3. Structural Classification of Sentences

- **Simple Sentences:** A simple sentence contains only one independent judgment. The sentences in (2), (3), (4), (5), and (6) are all examples of simple sentences.
- **Complex Sentences :** Complex sentences are those that include embedded dependent (subordinate) clauses (with verbal heads of kind infinitive, nominalization, participle or gerund) as their constituents, or as modifiers of their constituents. Dependent clauses may themselves contain other dependent clauses, resulting in embedded structures like (7):

(7) **Bura+da** **iç+il+ebil+ecek** **su**
here+LOC drink+PASS+POT +FUT–PART water
bul+ama+yacağ+ım+ı **zannet+mek** **doğru**
find+NEG–POT+FACT–NOM +1SG–POSS+ACC think+INF right
ol+maz+dı.
be+NEG–AOR+PAST(+3SG)
'It wouldn't be right to think that I wouldn't be able to find drinkable water here.'

The subject of (7) (*burada içilebilecek su bulamayacağımı zannetmek – to think that I wouldn't be able to find drinkable water here*) is an infinitival dependent clause whose accusative object (*burada içilebilecek su bulamayacağımı – that I wouldn't be able to find drinkable water here*) is a nominalization clause. The nominative object of this accusative object (*içilebilecek su – drinkable water*) is a noun phrase where the head noun (*su – water*) is modified by a participle clause (*içilebilecek – drinkable*).

It should be noted that there are other types of sentences in the classification according to structure, for which we will not provide any examples here because of space limitations. (See Şimşek [4], and Güngördü [8] for details.)

4. System Architecture and Implementation

We have implemented our parser in the grammar development environment of the Generalized LR Parser/Compiler (henceforth Parser/Compiler) developed at Carnegie Mellon University Center for Machine Translation. We have incorporated a separate morphological analyzer for handling Turkish morphology. The parser consists of four blocks as shown in Figure 1.

1. The block labeled *Turkish LFG Parser* is the overall control module which interfaces with the user and handles input and output, and invokes the other modules.
2. The morphological analyzer module implements a full scale two-level specification of Turkish morphology, covering all morpho-phonological aspects of the language. It uses the lexical database in the lexicon module. The morphological analyzer returns a list of *feature–value* tuples.[9]
3. The lexicon module has about 24,000 Turkish root words and is used for morphological analysis and parsing. For use in parsing, it has limited additional syntactic and semantic information (such as whether nouns indicate various kinds of temporal concepts, or materials, or whether they may act as containers, or whether adjectives are gradable, qualitative, the subcategorization requirements of postpositions, etc.) There is a separate verb lexicon of about 185 common verbs containing information about the subcategorization frames, each frame containing the grammatical functions, case features, optionality and thematic roles of the arguments involved.
4. The actual parsing is done by the parser module on the right, which is loaded with the LR parser tables generated by the grammar compiler from the grammar source.

When a sentence is given as input to the program, the program first calls the morphological analyzer for each word in the sentence, and keeps the results of these calls in a list to be used later by the parser.[10] If the morphological analyzer fails to return a structure for a word for any reason (e.g., the lexicon may lack the word or the word may be misspelled), the program returns with an error message. After the morphological analysis is completed, the parser is invoked to check whether the sentence is grammatical. The parser performs bottom-up parsing. During this analysis, whenever it consumes a new word from the sentence, it picks up the morphological structure of this word from the list. If the word is a verb (finite or non-finite), the parser is also provided with the subcategorization frame of the

[9] For instance, for the word *evdekilerin* (of those (things) in the house/your things in the house) the morphological analyzer returns:
(a) ((*CAT* N)(*R* "ev")(*CASE* LOC)(*CONV* ADJ "ki")(*AGR* 3PL)(*CASE* GEN))
(b) ((*CAT* N)(*R* "ev")(*CASE* LOC)(*CONV* ADJ "ki")(*AGR* 3PL)(*POSS* 2SG)).

[10] Recall that there may be a number of morphologically ambiguous interpretations of a word. In such cases, the morphological analyzer returns all of the possible morphological structures in a list, and the parser takes care of the ambiguity within the rules.

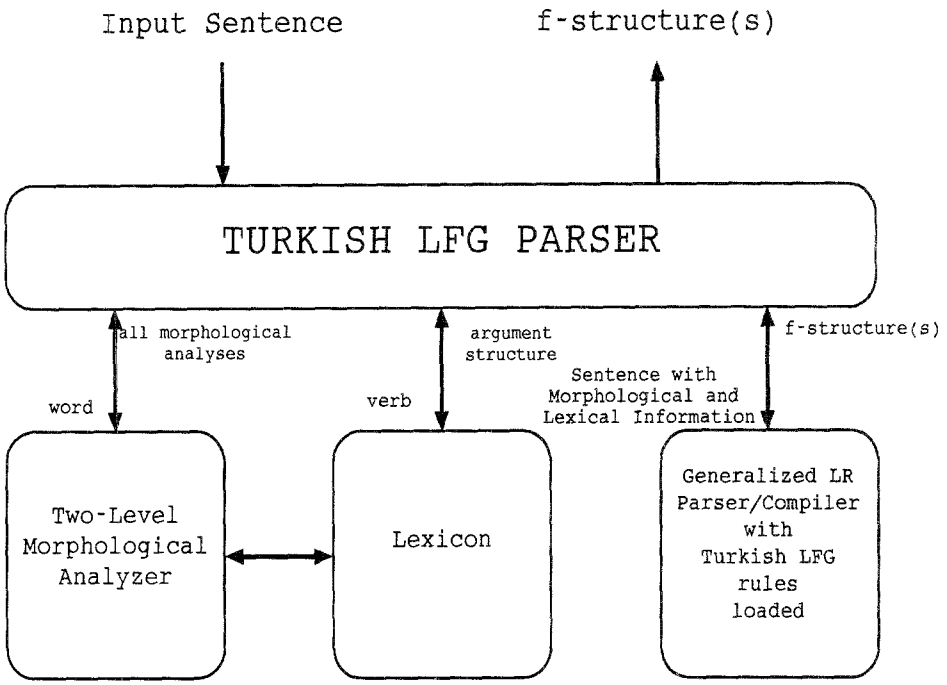

Fig. 1. The system architecture.

word. At the end of the analysis, if the sentence is grammatical, its f-structure is output by the parser.

5. The Grammar

In this section, we present an overview of the grammar component of our system. The grammar includes rules for *sentences, dependent clauses, noun phrases, adjectival phrases, postpositional phrases, adverbial constructs, verb phrases*, and a number of *lexical look up rules.*[11] Table I presents the number of rules for each category in the grammar.[12] There are also some intermediary rules, not shown here.

Recall from Section 3.2 that Turkish sentences have a rather free constituent order as a result of the highly inflected morphology. In other words, grammatical functions of the constituents in a sentence cannot be determined by relying on their order, but rather on their case features. The fact that in LFG information about

[11] Recall that no morphological rules have been included in the grammar. Instead there is one lexical look up rule for each lexical category, whose bare function is to call the morphological analyzer.

[12] Note that these figures belong to the actual implementation of the grammar on the Parser/Compiler, which does not support optionality or operations like Kleene closure in the phrase structure parts of the LFG rules.

TABLE I. The number of rules for each category in the grammar.

Category	Number of Rules
Noun phrases	17
Adjectival phrases	10
Postpositional phrases	24
Adverbial constructs	50
Verbal phrases	21
Dependent clauses	14
Sentences	6
Lexical look up rules	11
TOTAL	153

the grammatical functions in a sentence and information about its phrase structure are encoded in two different levels of syntactic structure (f- and c-structures, respectively) helps us to deal with this feature of Turkish syntax.

In our grammar, we assume a flat structure for Turkish sentences; that is, the subject and the other arguments and modifiers are attached to the verb at the same level. An alternative solution would be to assume a hierarchical sentence structure as in the case of English, and to deal with word order variation using a stylistic scrambling rule that acts on phrase structure rules. However, such an approach would lead to a big gap between the theoretical analysis and the actual implementation of the grammar since the Parser/Compiler does not support such operations on phrase structure rules.[13] Therefore, we assume that the following phrase structure rule is responsible for the derivation of sentence structures in Turkish:

(8) $S \rightarrow XP^* \ V \ XP^*$

The rule in (8) expands S into an arbitrary number of constituents at the XP level (NP, PP, ADVP, etc.) followed by a verb which is again followed by an arbitrary number of constituents at the XP level.[14]

[13] We discuss an additional reason for taking the former approach below, which relates to our generalization of freedom of word order in Turkish.

[14] Mohanan [18] proposes a similar analysis for Malayalam, which exhibits properties similar to Turkish in terms of word order. The sentence structure is flat (rather than hierarchical) in his analysis as well. However, he assumes that Malayalam is a verb-final language and lets a scrambling rule deal with the freedom of the verb to appear nonfinally. He discusses a number of syntactic, semantic and phonological features of Malayalam, which favor such an analysis. Although some of these "syntactic" properties hold true for Turkish as well (e.g., the existence of postpositions rather than prepositions, or the fact that auxiliaries always follow the main verbs), we prefer to stick to the phrase structure rule in (8), partly because we avoid the use of scrambling rules in the analysis as explained above.

Recall from Section 3.2 that a nominative direct object should be placed immediately before the verb, and that nonderived manner adverbs always immediately precede the verb or, if it exists, the nominative direct object. In our grammar, we treat such objects and adverbials as part of the verb phrase, revising (8) as follows:[15]

(9) a. S → XP* V′ XP*
b. V′ → (ADV) (NP) V

Such an approach enables us to generalize the word order variation in Turkish sentences by permitting freedom in the order of the sister constituents directly dominated by S. Hence, we do not need to check the constraints above in the sentence rule.[16]

Figure 2 shows an example grammar rule from the implementation, which deals with sentences with two constituents,[17] with an informal description of the equations part.[18,19]

There are a few points that require further clarification in this rule. We assume that NPs are assigned their case features in the lexicon through word formation rules. (Recall that this function is performed by the morphological analyzer in the implementation.) These case features are then used in determining grammatical functions of NPs, in the equations parts of the sentence rules (cf. the second and third "if-statements" of item (2) in Figure 2). Notice that we make a two way distinction here: If the case feature of the NP is nominative then it can be the subject.[20] Otherwise, it can be an object, in which case its type (i.e., direct, indirect, etc.) is determined using the subcategorization information associated with the verb, which contains information about the grammatical functions, case features, thematic roles and optionality of its arguments. Mohanan [18] proposes a set of "principles of case interpretation" for Malayalam that assign nominal expressions to the argument positions of the verb by interpreting their case features (assigned in the lexicon by the word formation rules). A similar set of principles

[15] Note that functional schemata associated with these phrase structure rules have been omitted, which, for example, in the case of (9b) check whether the ADV is a nonderived manner adverb and whether the case feature of the NP is nominative.

[16] There are similar sets of rules for dependent clauses as well. For participle clauses, which are always head-final, (9a) is replaced by the following:

(i) PartP → XP* Part′

[17] Since the Parser/Compiler does not support Kleene closure operation on phrase structure parts of the rules, we need to have separate rules to cover sentences with different number of constituents.

[18] A Parser/Compiler rule is composed of a phrase structure component and an equations component (which corresponds to the functional schemata associated with a phrase structure rule in LFG terminology).

[19] Note that *x0*, *x1*, and *x2* refer to the functional structures of the sentence, the first constituent and the second constituent in the phrase structure, respectively.

[20] Recall that nominative direct objects are attached to the verb by the V′ rule, rather than the sentence rule. Hence, a nominative NP can only be interpreted as subject at this level.

```
(<S> <==> (<XP> <XP>)
  1) if x1's category is VP then
        assign x1 to the functional structure of the
                  verb of the sentence
     if x2's category is VP then
        assign x2 to the functional structure of the
                  verb of the sentence

  2) for i = 1 to 2 do
     (use if, not else if, since there may be ambiguous parses)
       if xi has already been assigned to the functional
        structure of the verb then  do nothing

       if xi's category is ADVP then
          add xi to the adverbial adjuncts of the sentence

       if xi's category is NP and xi's case is nominative then
          assign xi to the functional structure of the
                subject of the  sentence

       if xi's category is NP and xi's case is not nominative then
          (coherence check)
          if the verb of the sentence can take an object with
             this case  (considering also the voice of the verb)
             add xi to the objects of the verb

     (completeness check)
  3) check if the verb has taken all the objects that it has
           to take

     (coherence check)
  4) make sure that the verb has not taken more than one object
       with the same grammatical role

  5) check if the subject and the verb agree in
           number and person:
        if the subject is defined (overt) then
         if the agreement feature of the subject is
              third  person plural then
             the agreement feature of the verb may be either
                 third person singular or third person plural

          else
             the agreement features of the subject and
                 the verb must be the same

        else if the subject is undefined (covert) then
          assign the agreement feature of the verb
           to that of the subject
```

Fig. 2. A sentence rule given with an informal description of the equations part.

can be suggested for Turkish as well. In this case, one would need to specify the case features of the arguments within the subcategorization information only in

some exceptional cases.[21]

The Parser/Compiler lets us make calls to ordinary LISP functions in the equations parts of the rules. So, we check the coherence and completeness of the f-structures assigned to sentences and embedded dependent clauses in the equations parts of the rules for these constructions. This is essentially equivalent to recursively checking the well-formedness at the end of the parse except that the checking for an embedded clause is done at the end of its parse, and not at the end of the parse of the matrix clause. Intuitively, one would expect this strategy to be more efficient than recursive checking at the end of the parse since spurious ambiguities that arise as a result of attaching arguments at one clause level to an embedded clause (or vice versa) would be eliminated at the end of the parse of the embedded clause rather than the whole sentence. Consider (10) as an example:

(10) **Ben** [**anne+m+e** [**Güneş+e** **dün**
I mother+1SG-POSS+DAT Güneş+DAT yesterday

okul+da **kitab+ı** **ver+diğ+im+i**]
school+LOC book+ACC give+FACT–NOM+1SG–POSS+ACC

söyle+me+yi] **unut+tu+m.**
tell+ACT–NOM+ACC forget+PAST+1SG

'I forgot to tell my mother that I gave the book to Güneş at the school yesterday.'

Note that in (10) there are five NPs and an adverb that could be attached to the following two nominalizations and a finite verb in any partition (provided that the resulting structure is nested) resulting in spurious ambiguities that would hang around till the end of the parse, in the absence of dynamic coherence and completeness checking.[22]

The final point to note about the rule in Figure 2 is the way subject-verb agreement is checked in item (5). Since Turkish is a pro-drop language, the subject of a sentence may be either overt or covert. If it is an overt subject, then the agreement features of the subject and the verb must agree with each other in the way defined in footnote 3. In the case of a covert subject, on the other hand, the agreement feature of the subject is simply unified with that of the verb.

[21] For example, there is a class of verbs in Turkish that idiosyncratically require their direct objects to be in dative or ablative case, rather than accusative. This property cannot be correlated with thematic roles. For example, *döv- 'beat'* and *sev- 'love'* take accusative direct objects, whereas *vur 'hit'* takes a dative and *hoşlan- 'like'* an ablative one.

[22] Needless to say, one can easily produce examples with greater number of NPs and adverbials and/or level of embedding.

In Section 3.2 we mentioned that word order in Turkish was also constrained by particular discourse conditions. In this work we do not take these constraints into consideration. King [14, chapters 6–8] provides an LFG account for the interaction between phrase structure, discourse functions, and grammatical relations in Russian. She suggests a hierarchical phrase structure, where certain positions are associated with particular discourse functions, to account for the interaction between word order and discourse function interpretation in Russian. In addition, the distribution of grammatical functions in the phrase structure is governed by the interaction of *functional uncertainty* [12] with the well-formedness conditions on the f-structure. Although discourse function information is represented in the f-structure in her account, she also discusses alternative solutions such as locating this information in the semantic-structure.

6. Performance Evaluation

In this section, we present some results about the performance of our system on test runs with four different texts on different topics. All of the texts are articles taken from magazines. We used the CMU Common Lisp system running in a Unix environment on SUN Sparcstations at Centre for Cognitive Science, University of Edinburgh.[23]

In all of the texts there were some sentences outside our scope. These were:

- sentences with finite sentences as their constituents or modifiers of their constituents,
- conditional sentences,
- finite sentences that were connected by conjunctions, and
- sentences with discontinuous constituents.[24]

We pre-edited the texts so that the sentences were in our scope (e.g., separated finite sentences connected by conjunctions and commas, and parsed them as independent sentences, and ignored the conditional sentences). Table II presents some statistical information about the test runs. The first, second and third columns show the document number, the total number of sentences and the number of sentences that we could parse without pre-editing, respectively. The other columns show the number of sentences that we totally ignored, the number of sentences in the pre-edited versions of the documents, average number of parses per sentence generated and average runtime for each of the sentences in the texts, respectively. It can be seen that our grammar can successfully deal with about 82% of the sentences that we have experimented with, with almost all the remaining sentences becoming parsable after a minor pre-editing. This indicates that our grammar coverage is

[23] We should, however, note that the times reported are exclusive of the time taken by the morphological analyzer, which, with a 24,000 word root lexicon, is rather slow and can process about 2 lexical forms per second.

[24] Again, this is a consequence of the word order freeness in Turkish.

TABLE II. Statistical information about the test runs.

Text	Number of Sentences	Sentences in Scope	Sent. ignored	Sent. after Pre-editing	Avg. Parses per Sentence	Avg. CPU Time per Sentence
1	43	30	0	55	4.28	12.26 sec.
2	51	41	2	62	5.02	8.92 sec.
3	56	48	1	64	4.87	10.28 sec.
4	80	70	0	97	3.25	7.46 sec.
Total	230	189(82%)	3	279	–	–

TABLE III. Impact of disambiguation on parsing performance

Avg. Length (words)	No disambiguation		With disambiguation		Ratios	
	Avg. parses	Avg. time (sec)	Avg. parses	Avg. time (sec)	parses	speed-up
5.7	5.78	29.11	3.30	11.91	1.97	2.38

Note: The ratios are the averages of the sentence by sentence ratios.

reasonably satisfactory, at least for the texts concerned, which, incidentally, were texts from magazines.

In languages like Turkish with words that are morphologically ambiguous due to ambiguities in the part-of-speech of the root, or to different ways of interpreting the suffixes, using a tagger that relies on various sources of information (contextual constraints, usage statistics, lexical preferences and heuristics) to preprocess the input, can have a significant impact on parsing. We have tested the impact of morphological and lexical disambiguation on the performance of the parser by tagging our input using the tagger that we have developed in a different work [10, 23]. This tagger was not an integrated part of the system architecture proper, but was an off-line system which did morphological analysis and disambiguation on the sentences which were then passed to the parser, which in this case skipped over the morphological analysis phase. The results were compared to the case when the parser had to consider all possible morphological ambiguities itself. For a set of 80 sentences considered, it can be seen in Table III that morphological disambiguation enables almost a factor of two reduction in the average number of parses generated and over a factor of two speed-up in time.[25]

[25] This set of measurements were performed on a slower machine and hence the differences in absolute parsing time.

7. Example Outputs

In this section we provide a number of examples that highlight the capabilities of our parser. The first example we present is for a sentence which shows very nicely where the structural ambiguity comes out in Turkish.[26] The output for (11a) indicates that there are four ambiguous interpretations for this sentence as indicated in (11b–e):[27]

(11a)	**Küçük**	**kırmızı**	**top**	**git+tikçe**	**hızlan+dı.**
	little	red	ball	go+GER	speed up+PAST(+3SG)
		kırmız+ı		gradually	
		red paint/insect			
		+3SG–POSS			

(11b) 'The little red ball gradually sped up.'

(11c) 'The little red (one) sped up as the ball went.'

(11d) 'The little (one) sped up as the red ball went.'

(11e) 'It sped up as the little red ball went.'

The output of the parser for the first interpretation, which is in fact semantically the most plausible one, is given in Figure 3.[28,29] This output indicates that the subject of the sentence is a noun phrase whose modifier part is *küçük*, and modified part is another noun phrase whose modifier part is *kırmızı* and modified part is *top*. The agreement of the subject is third person singular, case is nominative, etc. *Hızlandı* is the verb of the sentence, and its voice is active, tense is past, agreement is third person singular, etc. *Gittikçe* is a temporal adverbial adjunct, derived from the verbal root *git* with the suffix *dikçe*.

Figures 4 through 7 illustrate the c-structures of the four ambiguous interpretations (11b–e), respectively.[30] Note that:

– In (11b), the adjective *kırmızı* modifies the noun *top*, and this noun phrase is then modified by the adjective *küçük*. The entire noun phrase functions as

[26] This example is not in any of the texts mentioned above. It is taken from the first author's M.Sc. thesis [8].

[27] In fact, there is also a fifth interpretation due to the lexical ambiguity of the second word. In Turkish, *kırmız* is the name of a shining, red paint obtained from an insect with the same name. So, (11a) also means '*His little red paint/insect sped up as the ball went.*' However, this is very unlikely to come to mind even for native speakers.

[28] The system uses upper case ASCII characters to represent special characters of the Turkish alphabet, e.g., ü is represented with U.

[29] The other parses may conceivably, albeit very improbably, be used in certain discourse situations. As native speakers, we feel that the semantics expressed by these other parses would probably be expressed differently.

[30] The c-structures given here are simplified by removing some nodes introduced by certain intermediary rules, to increase readability.

```
((SUBJ
   ((*AGR* 3SG) (*CASE* NOM)
                (*DEF* NIL)
                (*CAT* NP)
                (MODIFIED
                  ((*CAT* NP)
                    (MODIFIER
                      ((*CASE* NOM) (*AGR* 3SG)
                                    (*LEX* "kIrmIzI")
                                    (*CAT* ADJ)
                                    (*R* "kIrmIzI")))
                    (MODIFIED
                      ((*CAT* N) (*CASE* NOM)
                                 (*AGR* 3SG)
                                 (*LEX* "top")
                                 (*R* "top")))
                    (*AGR* 3SG)
                    (*CASE* NOM)
                    (*LEX* "top")
                    (*DEF* NIL)))
                (MODIFIER
                  ((*SUB* QUAL) (*CASE* NOM)
                                (*AGR* 3SG)
                                (*CAT* ADJ)
                                (*LEX* "kUCUk")))
                (*LEX* "top")))
 (VERB
      ((*TYPE* VERBAL) (*VOICE* ACT)
                       (*LEX* "hIzlandI")
                       (*CAT* V)
                       (*R* "hIzlan")
                       (*TENSE* PAST)
                       (*AGR* 3SG)))
 (ADVADJUNCTS
      ((*SUB* TEMP) (*LEX* "gittikCe")
                    (*CAT* ADVP)
                    (*CONV*
                      ((*WITH-SUFFIX* "dikce") (*CAT* V)
                                               (*R* "git"))))))
```

Fig. 3. Output of the parser for the first the ambiguous interpretation of (11a) (i.e., (11b)).

the subject of the main verb *hızlandı*, and the gerund *gittikçe* functions as an adverbial adjunct of the main verb.

- In (11c), the adjective *kırmızı* is used as a noun, and is modified by the adjective *küçük*.[31] This noun phrase functions as the subject of the main verb. The noun *top* functions as the subject of the gerund *gittikçe*, and this gerundive clause functions as an adverbial adjunct of the main verb.
- In (11d), the adjective *küçük* is used as a noun, and functions as the subject of the main verb. The noun phrase *kırmızı top* functions as the subject of the

[31] In Turkish, any adjective can be used as a noun.

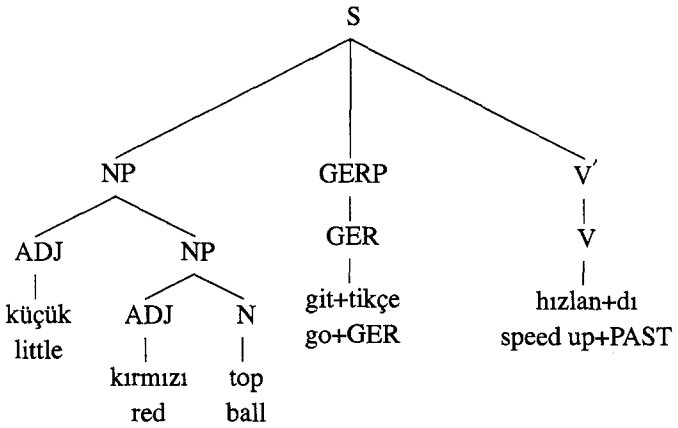

Fig. 4. C-structure for (11b).

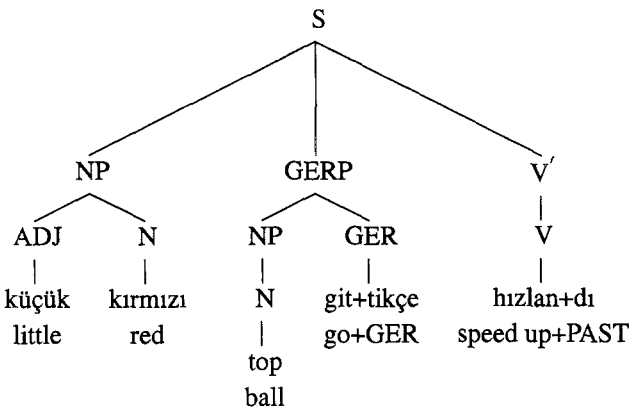

Fig. 5. C-structure for (11c).

gerund *gittikçe*, and this gerundive clause functions as an adverbial adjunct of the main verb.

– Finally, in (11e), the noun phrase *küçük kırmızı top* functions as the subject of the gerund *gittikçe* (cf. (11b), where it functions as the subject of the main verb), and this gerundive clause functions as an adverbial adjunct of the main verb. Note that the subject of the main verb in this interpretation (i.e., *it*) is a covert one. Hence, it does not appear in the c-structure shown in Figure 7.

It can be seen that the ambiguities result essentially from the various ways the initial noun phrase can be apportioned into two separate noun phrases, one being the subject of the main sentence, and the other being the subject of the embedded gerundive clause. This is possible in this case since all Turkish adjectives can function as nouns effectively modifying a covert third person singular nominal. It is possible to rank these ambiguities in a post-processing stage where, for example,

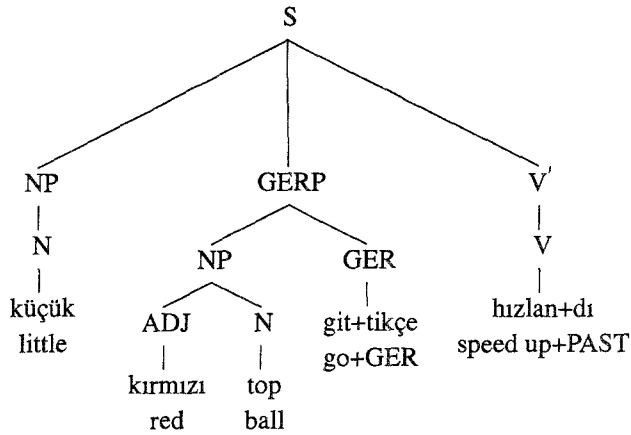

Fig. 6. C-structure for (11d).

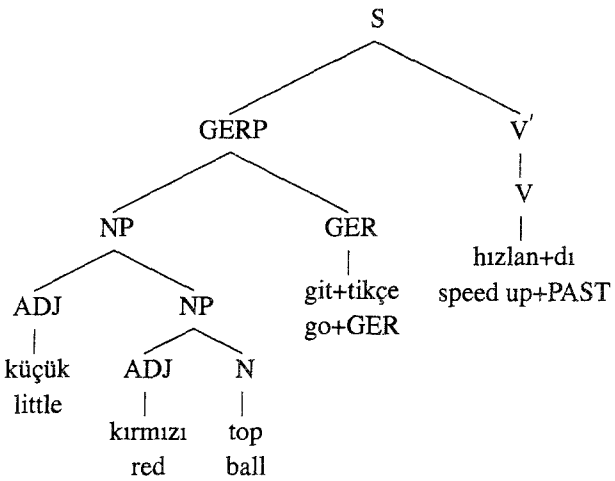

Fig. 7. C-structure for (11e).

parses with the longest noun phrases and/or with an overt subject in the main clause are preferred.

The second example is for a rather complicated sentence (7) given earlier, which involves embedded dependent clauses. We repeat it here for convenience:

(7)	**Bura+da**	**iç+il+ebil+ecek**	**su**
	here+LOC	drink+PASS+POT +FUT–PART	water
	bul+ama+yacağ+ım+ı	**zannet+mek**	**doğru**
	find+NEG–POT+FACT–NOM +1SG–POSS+ACC	think+INF	right

ol+maz+dı.
be+NEG–AOR +PAST(+3SG)

'It wouldn't be right to think that I wouldn't be able to find drinkable water here.'

Figure 8 shows the c-structure for the intended interpretation.[32] Although the gloss above is the intended or preferred interpretation of this sentence, where the locative adjunct *burada* is attached to the participle phrase *içilebilecek su bulamayacağımı*, the parser generates additional parses which attach *burada* to each of the other two embedded clauses and the main verb, resulting in three more parses:[33]

1. It would not be right to think that I would not be able find water that could not be drunk *here* (literally – not drinkable here) (where *burada* modifies the participle *içilebilecek*).
2. It would not be right to think *here* that I would not be able find drinkable water (where *burada* modifies the infinitive *zannetmek*).
3. It would not be right *here*, to think that I would not be able to find drinkable water (where *burada* modifies the main verb *olmazdı*).

This example shows another aspect of Turkish syntax that we deal with in a very limited fashion (though not in this specific example): that of using punctuation information to resolve attachment ambiguities. For instance, a comma after the locative adjunct *burada* would attach it to the main verb *olmazdı* corresponding to the third interpretation above, while the lack of this comma could be taken as a basis to rule out this interpretation.

The third example that we present serves to emphasize our capability in dealing with word order freeness. Our approach to handling word order freeness does not deal with all of the subtle issues involved. We accept a sentence to be grammatically correct if the order of the constituents (at every level) does not violate certain constraints (namely, those that we discuss in Section 3.2) and if its f-structure satisfies the well-formedness conditions presented in Section 2.

The example is the following sentence:

(12)	**Ben**	**kitab+ı**	**ev+den**	**okul+a**	**götür+dü+m.**
	I	book+ACC	house+ABL	school+DAT	take+PAST+1SG

'I took the book from the house to the school.'

[32] We have opted not to present the f-structure as it does not provide any additional insights. See Güngördü and Oflazer [9] for the details of the f-structure of this parse.

[33] In fact, a number of other parses are generated due to the fact that the nominalization *bulamayacağımı* can be interpreted as a nominal phrase on its own. This is because although the root verb *bul-* (find) is transitive, its object is optional (which is true of almost all Turkish transitive verbs). In this case the preceding noun phrase *içilebilecek su* is not attached as an object noun phrase to this nominalization, but rather acts as a modifier for its nominal interpretation, resulting in a syntactically valid nominal compound.

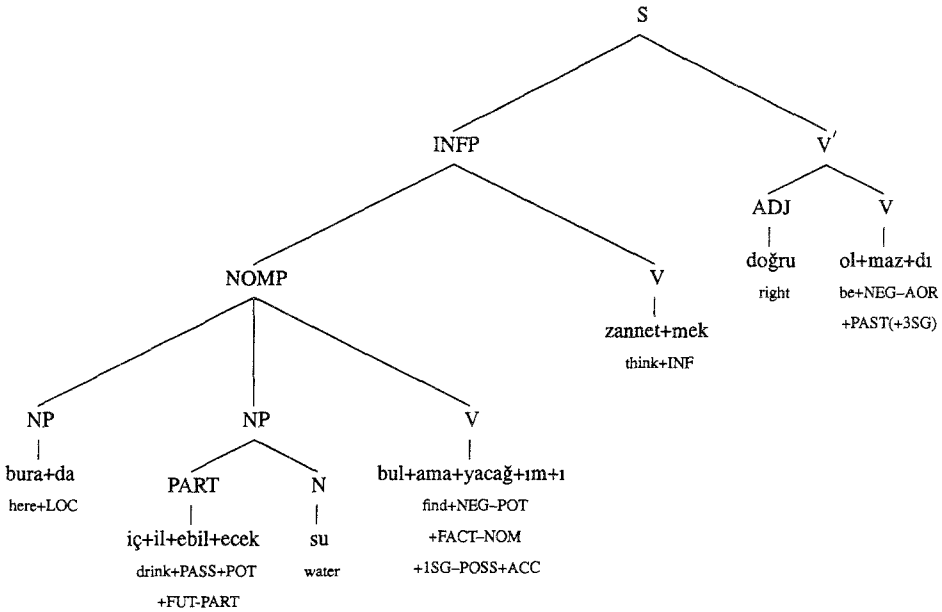

Fig. 8. C-structure for the intended interpretation of (7).

Our system processes this as follows:

```
Enter the sentence :  ben kitabI evden okula  gOtUrdUm

("ben" "kitabI" "evden" "okula" "gOtUrdUm")
Total time in Morphological Analyzer =      736 Msecs
Avg/word =      147 Msecs

((((*LEX* "ben")(*CAT* N)(*R* "ben")(*AGR* 3SG)(*CASE* NOM))
  ((*LEX* "ben")(*CAT* PN)(*R* "ben") (*AGR 1SG)(*CASE* NOM)))
 (((*LEX* "kitabI")(*CAT* N)(*R* "kitap")(*AGR* 3SG)(*POSS* 3SG))
  ((*LEX* "kitabI")(*CAT* N)(*R* "kitap")(*AGR* 3SG)(*CASE* ACC)))
 (((*LEX* "evden")(*CAT* N)(*R* "ev")(*AGR* 3SG)(*CASE* ABL)))
 (((*LEX* "okula")(*CAT* N)(*R* "okul")(*AGR* 3SG)(*CASE* DAT)))
 (((*LEX* "gOtUrdUm")(*CAT* V)(*R* "gOtUr")(*TENSE* PAST)
                                          (*AGR* 1SG))))

1 (1) ambiguity found and took 2.454042 seconds of real time
```

The functional structure that is output for this case is the following:

```
;**** ambiguity 1 ***
((SUBJ
   ((*AGR* 1SG) (*CASE* NOM)
                (*CAT* NP)
                (*DEF* +)
                (*LEX* "ben")
                (*R* "ben")))
 (OBJS
     (*MULTIPLE*
         ((*CASE* DAT) (*R* "okul")
```

```
                        (*LEX* "okula")
                        (*AGR* 3SG)
                        (*DEF* NIL)
                        (*CAT* NP)
                        (*TYPE* OBLIQUE)
                        (*ROLE* GOAL))
            ((*CASE* ABL) (*R* "ev")
                        (*LEX* "evden")
                        (*AGR* 3SG)
                        (*DEF* NIL)
                        (*CAT* NP)
                        (*TYPE* OBLIQUE)
                        (*ROLE* SOURCE))
            ((*CASE* ACC) (*DEF* +)
                        (*R* "kitap")
                        (*LEX* "kitabI")
                        (*AGR* 3SG)
                        (*CAT* NP)
                        (*TYPE* DIRECT)
                        (*ROLE* THEME))))
    (VERB
        ((*CAT* V)
         (*TYPE* VERBAL)
         (*VOICE* ACT)
         (ARGS
            (((*CASE* (NOM ACC)) (*TYPE* DIRECT)
                                 (*OCC* OPTIONAL)
                                 (*ROLE* THEME))
             ((*CASE* DAT) (*TYPE* OBLIQUE)
                           (*OCC* OPTIONAL)
                           (*ROLE* GOAL))
             ((*CASE* ABL) (*TYPE* OBLIQUE)
                           (*OCC* OPTIONAL)
                           (*ROLE* SOURCE))))
         (*LEX* "gOtUrdUm")
         (*R* "gOtUr")
         (*TENSE* PAST)
         (*AGR* 1SG))))
```

Note that at this point we are not able to extract discourse-related information like topic, focus, background information, which is mostly marked using the constituent order in Turkish [6].

The following summary of outputs shows what our approach can handle in terms of word-order freeness. All valid parses produce the same functional structure:[34]

[34] For the first sentence, there is a syntactically correct second interpretation due to the lexical ambiguity of the word *ben* (pronoun *I*, or noun *mole*). The second interpretation when followed by a noun with the compound marker (CM) (*kitabı* – whose surface form is the same as that of the accusative form of the root *kitap*) forms a syntactically valid nominal compound *ben kitabı*, in which case the subject of the whole sentence is assumed to be covert and just marked with the agreement suffix in the verb:

(i)	**Ev+den**	**okul+a**	**ben**	**kitab+ı**	**götür+dü+m.**
	house+ABL	school+DAT	I	book+ACC	take+PAST+1SG
			mole	book+CM	

'I took the book from the house to the school.'

'I took a mole book from the house to the school.'

```
Enter the sentence:  evden okula ben kitabI gOtUrdUm
 2 (2) ambiguities found and took 2.128624 seconds of real time

Enter the sentence :  evden ben okula kitabI gOtUrdUm
 1 (1) ambiguity found and took 1.650397 seconds of real time

Enter the sentence :  evden kitabI okula ben gOtUrdUm
 1 (1) ambiguity found and took 1.906963 seconds of real time

Enter the sentence :  okula evden kitabI ben gOtUrdUm
 1 (1) ambiguity found and took 1.749944 seconds of real time

Enter the sentence :  okula kitabI ben evden gOtUrdUm
 1 (1) ambiguity found and took 2.176758 seconds of real time

Enter the sentence :  evden kitabI ben okula gOtUrdUm
 1 (1) ambiguity found and took 1.713014 seconds of real time

Enter the sentence :  kitabI okula ben evden gOtUrdUm
 1 (1) ambiguity found and took 1.842986 seconds of real time

Enter the sentence :  gOtUrdUm ben okula evden kitabI
 1 (1) ambiguity found and took 1.439124 seconds of real time

Enter the sentence :  okula gOtUrdUm ben evden kitabI
 1 (1) ambiguity found and took 1.370975 seconds of real time

Enter the sentence :  ben kitap gOtUrdUm evden okula
 1 (1) ambiguity found and took 1.487312 seconds of real time

Enter the sentence :  kitap ben gOtUrdUm evden okula
 failed

Enter the sentence :  ben kitap evden okula gOtUrdUm
 failed
```

The last two examples in this summary display cases where the position of the nominative direct object *kitap* has strayed from the immediately preverbal position rendering these sentences ungrammatical (cf. the constraint on nominative direct objects given in Section 3.2).

Finally, consider the following example regarding the constraints on word order mentioned in Section 3.2. In the case of (13), the parser generates two ambiguities where, in the first one the adjective *hızlı* modifies the succeeding noun *araba*, and in the second one it acts as an adverbial adjunct modifying the verb *götürdüm*:

(13) **Ben ev+den okul+a hızlı araba götür+dü+m.**
I house+ABL school+DAT fast car take+PAST+1SG
'I took a fast car from the house to the school.'
'I quickly took a car from the house to the school.'

```
Enter the sentence :  ben evden okula hIzlI araba gOtUrdUm

("ben" "evden" "okula" "hIzlI" "araba" "gOtUrdUm")
```

```
Total time in Morphological Analyzer =      925 Msecs
Avg/word =      154 Msecs
....
2 (2) ambiguities found and took 5.820933 seconds of real time
```

If, however, *hızlı* appears in the immediately preverbal position, the sentence becomes ungrammatical and is rejected by the parser since the nominative direct object *araba* does not immediately precede the verb:

```
Enter the sentence :  ben evden okula araba hIzlI gOtUrdUm

("ben" "evden" "okula" "araba" "hIzlI" "gOtUrdUm")
Total time in Morphological Analyzer =      880 Msecs
Avg/word =      146 Msecs

failed
```

On the other hand, had the direct object *araba* been accusative (with the surface form *arabayı*) then we would have a grammatical sentence even if the adverb were immediately preverbal:

(14) **Ben ev+den okul+a araba+yı hızlı götür+dü+m.**
I house+ABL school+DAT car+ACC fast take+PAST+1SG
'I quickly took the car from the house to the school.'

```
Enter the sentence :  ben evden okula arabayI hIzlI gOtUrdUm

("ben" "evden" "okula" "arabayI" "hIzlI" "gOtUrdUm")
Total time in Morphological Analyzer =      871 Msecs
Avg/word =      145 Msecs
 1 (1) ambiguity found and took 2.938792 seconds of real time
.....
```

8. Discussion and Conclusions

We have presented a summary and highlights of our current work on parsing Turkish using a unification-based computational framework. This is the first such effort for constructing a unification-based parser for Turkish with such a wide coverage. The parser has been implemented using the Generalized LR Parser/Compiler developed at Carnegie Mellon University Center for Machine Translation. The morphological analysis of Turkish lexical forms are handled by incorporating a full-scale two-level morphological analyzer into the parser. Evaluations using well over 200 sentences from Turkish magazine articles indicate that a large percentage of the sentences can be parsed directly and almost all the rest with minimal pre-editing. Our grammar covers structurally simple and complex declarative sentences and questions but does yet not deal with sentences involving finite sentences as their constituents or modifiers of their constituents, conditional sentences, sentences with discontinuous constituents, and sentences with coordinate finite clauses.

The work presented in this paper may serve two purposes in the context of machine translation. For a machine translation application where the source language is Turkish, this system may function as the source language analysis front-end, with the generation of the target language being performed by a separate system. It is certainly possible to utilize the LFG formalism to implement the transfer component of a transfer-based system along the lines of Kaplan and Wedekind[24] or as presented in van Eynde [30], if both the source and target language components are based on the LFG formalism.

The grammar itself may with reasonable ease be transformed into a grammar, that can be used by a generation system for Turkish, using the generation tool based on the same grammar formalism [29]. Such a grammar forms the basis of a target language generation system in a machine translation system where Turkish is the target language. We are currently working on constructing such a generation grammar based on the grammar presented in this paper as a part of our on-going work [22].

We have a number of directions for improving our grammar and parser, some of which are being undertaken again as a part of our on-going work:

- Turkish is very rich in terms of non-lexicalized collocations where a sequence of lexical forms with a certain set of morpho-syntactic constraints is interpreted from a syntactic point as a single entity with a completely different part of speech. For instance any sequence like:

  verb+AOR+3SG verb+NEG+AOR+3SG

  with both verbal roots the same, is equivalent to the manner adverbial "by verb+ing" in English, yet the relations between the original verbal root and its complements are still in effect. We currently deal with these in the parser, but our tagger [10, 23] can successfully deal with these and we expect to integrate this functionality to relieve the parser from dealing with such lexical problems at syntactic level.
- We are currently working on extending our coverage to make it cover the types of sentences other than structurally simple and complex ones as well.
- Turkish verbs have typically many idiomatic meanings when they are used with subjects, objects, adverbial adjuncts with certain lexical, morphological and semantic features. For example, the verb *ye-* (*eat*), when used with the object:
  - *para* (*money*) with no case and possessive marking, means to accept bribe,
  - *para* with obligatory accusative marking and optional possessive marking, means to spend money,
  - *kafa* (*head*) with obligatory accusative marking and no possessive marking, means to get mentally deranged,
  - *hak* (*right*) with optional accusative and possessive marking, means to be unfair to somebody,

- *baş* (*head*) (or a noun denoting a human) with obligatory accusative and possessive marking (obligatory only with *baş*), means to waste or demote a person.

Clearly such usage has impact on thematic role assignments to various role fillers, and even on the syntactic behavior of the verb in question. For instance, for the second and third cases, a passive form would not be grammatical. We have designed and built a verb lexicon and verb sense and idiomatic usage disambiguator [31] to deal with this aspect of Turkish explicitly and are in the process of integrating it into the parser. This verb lexicon is inspired by the CMU-CMT approach [16, 20] and in addition uses an ontological database represented in the LOOM [3] system for evaluating complex selectional constraints.

Acknowledgments

We would like to thank Carnegie-Mellon University, Center for Machine Translation for making available to us their LFG parsing system. We would also like to thank Elisabet Engdahl and Matt Crocker of Centre for Cognitive Science, University of Edinburgh, for providing valuable comments on an earlier version of this paper. The paper also benefited from the comments made by an anonymous referee who suggested comparisons with the works of King [14], and Mohanan [18]. This work was done as a part of a large scale NLP project and was supported in part by a NATO Science for Stability Grant TU-LANGUAGE.

References

1. Evan L. Antworth. *PC-KIMMO: A Two-level Processor for Morphological Analysis.* Summer Institute of Linguistics, 1990.
2. Joan Bresnan, editor. *The Mental Representation of Grammatical Relations.* MIT Press, 1982.
3. David Brill. *Loom User's Guide.* University of Southern California–Information Sciences Institute, 1992.
4. Rıza Şimşek. *Örneklerle Türkçe Sözdizimi (Turkish Syntax with Examples).* Kuzey Matbaacılık, 1987.
5. Mürvet Enç. The semantics of specificity. *Linguistic Inquiry*, 22(1):1–25, Winter 1991.
6. Eser E. Erguvanlı. *The Function of Word Order in Turkish Grammar.* PhD thesis, Department of Linguistics, University of California, Los Angeles, 1979.
7. Eser E. Erguvanlı. The role of semantic features in Turkish word order. *Folia Linguistica*, XXI:215–229, 1987.
8. Zelal Güngördü. A lexical-functional grammar for Turkish. Master's thesis, Department of Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey, July 1993.
9. Zelal Güngördü and Kemal Oflazer. Parsing Turkish using the Lexical-Functional Grammar formalism. In *Proceedings of COLING-94, the 15th International Conference on Computational Linguistics*, volume 1, pages 494–500, Kyoto, Japan, 1994.
10. Ilker Kuruöz. Tagging and morphological disambiguation of Turkish text. Master's thesis, Department of Computer Engineering and Information Sciences, Bilkent University, Ankara, Turkey, July 1994.
11. Ronald Kaplan and Joan Bresnan. *The Mental Representation of Grammatical Relations*, chapter Lexical-Functional Grammar: A Formal System for Grammatical Representation, pages 173–281. MIT Press, 1982.

12. Ronald Kaplan and Annie Zaenen. *Alternative Conceptions of Phrase Structure*, chapter Long-distance Dependencies, Consituent Structure, and Functional Uncertainty. IL: Chicago University Press, 1988.
13. Lauri Karttunen and Kenneth R. Beesley. Two-level rule compiler. Technical Report, XEROX Palo Alto Research Center, 1992.
14. Tracy H. King. *Configuring Topic and Focus in Russian.* PhD thesis, Department of Linguistics, Stanford University, 1993.
15. Robert H. Meskill. *A Transformational Analysis of Turkish Syntax.* Mouton, The Hague, Paris, 1970.
16. Ingrid Meyer, Boyan Onyshkevych, and Lynn Carlson. Lexicographic principles and design for knowledge-based machine translation. Technical Report CMU-CMT-90-118, Carnegie-Mellon University, Center for Machine Translation, 1990.
17. Teruko Mitamura, Hireyuki Musha, and Marion Kee. *The Generalized LR Parser/Compiler Version 8.1: User's Guide.* Carnegie-Mellon University – Center for Machine Translation, April 1988.
18. K.P. Mohanan. *The Mental Representation of Grammatical Relations*, chapter Grammatical Relations and Clause Structure in Malayalam, pages 504–589. MIT Press, 1982.
19. Birgit Nilsson. *Case Marking Semantics in Turkish.* PhD thesis, University of Stockholm, Stockholm, 1985.
20. Sergei Nirenburg, Jaime Carbonell, Masaru Tomita, and Kenneth Goodman. *Machine Translation: A Knowledge-based Approach.* Morgan Kaufman Publishers, 1992.
21. Kemal Oflazer. Two-level description of Turkish morphology. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, April 1993. A full version appears in *Literary and Linguistic Computing*, Vol. 9 No. 2, 1994.
22. Kemal Oflazer and Cem Bozşahin. Turkish Natural Language Processing Initiative: An overview. In *Proceedings of the Third Turkish Symposium on Artifical Intelligence.* Middle East Technical University, 1994.
23. Kemal Oflazer and İlker Kuruöz. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, pages 144–149. ACL, 1994.
24. Ronald Kaplan and Jürgen Wedekind. Restriction and correspondence-based translation. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 193–202, Utrecht, The Netherlands, April 1993.
25. Peter Sells. *Lectures on Contemporary Syntactic Theories.* CSLI-Lecture Notes 3, 1985.
26. Stuart M. Shieber. *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lecture Notes 4, 1986.
27. Albert M. Stoop. Atmaca: Semantic analysis by the computer. In Sabri Koç, editor, *Studies on Turkish Linguistics, Proceedings of the Fourth International Conference on Turkish Linguistics*, pages 539–564, Ankara, Turkey, 1988. Middle East Technical University.
28. Masaru Tomita. An efficient augmented-context-free parsing algorithm. *Computational Linguistics*, 13:31–46, January-June 1987.
29. Masaru Tomita and Eric H. Nyberg. *Generation Kit and Transformation Kit, Version 3.2 User's Manual.* Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA.
30. Frank van Eynde, editor. *Linguistic Issues in Machine Translation*, 1993.
31. Okan Yılmaz. Design and implementation of a verb lexicon and a verb sense disambiguator for Turkish. Master's thesis, Department of Computer Engineering and Information Science, Bilkent University, Ankara, Turkey, September 1994.