A Link Grammar for an Agglutinative Language

Ozlem Istek
Department of Computer Engineering
Bilkent University
Bilkent 06800, Ankara, Turkey
oistek@cs.bilkent.edu.tr

Ilyas Cicekli
Department of Computer Engineering
Bilkent University
Bilkent 06800, Ankara, Turkey
ilyas@cs.bilkent.edu.tr

Abstract

This paper presents a syntactic grammar developed in the link grammar formalism for Turkish which is an agglutinative language. In the link grammar formalism, the words of a sentence are linked with each other depending on their syntactic roles. Turkish has complex derivational and inflectional morphology, and derivational and inflection morphemes play important syntactic roles in the sentences. In order to develop a link grammar for Turkish, the lexical parts in the morphological representations of Turkish words are removed, and the links are created depending on the part of speech tags and inflectional morphemes in words. Furthermore, a derived word is separated at the derivational boundaries in order to treat each derivation morpheme as a special distinct word, and allow it to be linked with the rest of the sentence. The derivational morphemes of a word are also linked with each other with special links to indicate that they are parts of the same word. The adapted unique link grammar formalism for Turkish provides flexibility for the linkage construction, and similar methods can be used for other languages with complex morphology.

Keywords: parsing, link grammar.

1. Introduction

There are different classes of theories for the natural language syntactic parsing problem and for creating the related grammars. One of these classes of formalisms is categorical grammar motivated by the principle of compositionality. According to this formalism; syntactic constituents combine as functions or in a function-argument relationship. In addition to categorical grammars, there are two other classes of grammars, which are phrase structure grammars, and dependency grammars. Phrase structure grammars construct constituents in a tree-like hierarchy. On the other hand, dependency grammars build simple relations between pairs of words. Since dependency grammars are not defined by a specific word order, they are well suited to languages with free word order, such as Czech and Turkish. Link grammar [8] is similar to dependency grammar, but link grammar includes directionality in the relations between words, as well as lacking a head-dependent relationship.

There is some research on the computational analysis of Turkish syntax. One of these is a lexical functional grammar of Turkish [4]. There is also an ATN grammar for Turkish [2]. Another grammar for Turkish is based on HPSG formalism [9]. In addition, there are some works on the categorical grammars for Turkish [1,5]. Turkish syntax is also studied from the dependency parsing perspective. Oflazer presents a dependency parsing scheme using an extended finite state approach [6]. This parser is used for building a Turkish Treebank [7]. The Turkish Dependency Treebank is used for training and testing a statistical dependency parser for Turkish [3].

Syntactic analysis underlies most of the natural language applications and hence it is a very important step for any language. Although there are previous works on the computational analysis of Turkish, this paper presents the first link grammar developed for Turkish which is an agglutinative language. In this work, lexicalized structure of link grammar formalism is utilized for expressing the syntactic roles of intermediate derived forms of words in a language with very productive derivational and inflectional morphology. This is achieved by treating each of these intermediate derived forms as separate words. Using the adapted link grammar formalism, a fully functional link parser for Turkish is developed. The adapted link grammar formalism can also be used in the development of link grammars for other languages with very productive morphology.

Section 2 presents a general overview of the link grammar formalism, and Section 3 presents some distinctive features of Turkish syntax. In Section 4, the system architecture of the developed Turkish parser which uses our adapted link grammar formalism is given. Section 5 presents the special method for handling the syntactic roles of the words with derivations is given. Then, the paper continues with the performance evolution in Section 6, and Section 7 presents the concluding remarks.

2. Link Grammar

Link grammar is a formal grammatical system developed by Sleator and Temperley in 1993. In their work, they also developed top-down dynamic programming algorithms to process grammars based on this formalism and constructed a wide coverage link grammar for English. In this formalism, the syntax of a language is defined by a grammar that includes the words of the language and their linking requirements. A given sentence is accepted by the system if the linking requirements of all the words in the sentence are satisfied (connectivity), none of the links between the words cross each other (planarity) and there is at most one link between any pair of words (exclusion). A set of links between the words of a sentence that is accepted by the system is called a linkage. The grammar is defined in a dictionary file and each of the linking requirements of words is expressed in terms of connectors in the dictionary file. When a sequence of words is accepted, all the links are drawn above the words.

For example, the linkage requirements of three Turkish words can be defined as follows:

```
yedi (ate): O- & S-;
kadın (the woman): S+ ;
portakalı (the orange): O+;
```

Here, the verb "yedi"(ate) has two left linking requirements, one is "S"(subject) and the other is "O"(object). On the other hand, the noun "kadın" (the woman) needs to attach to a word on its right for its "S+" connector and the noun "portakalı"(the orange) has to attach a word on its right for its "O+" connector. Since the word, "yedi"(ate) and "kadın" (the woman) have the same "S" connector, i.e. same linking requirements, with opposite sign they can be connected by an "S" link. A similar situation occurs between the words "portakalı"(the orange) and "yedi"(ate) for the "O" connector. Therefore, if these words are connected in the following way, all of the linking requirements of these words are satisfied.

- Kadın portakalı yedi.
- (The woman ate the orange)

```
  +----------S----------+
  |          +----O-----+
  |          |          |
Kadın      portakalı   yedi
The woman  the orange  ate
```

In this sentence, "kadın"(the woman) links to word "yedi"(ate) with the S (subject) link and "portakalı"(the orange) links to word "yedi"(ate) with the O (object) link.

3. Turkish Syntax

In Turkish, the basic word order is SOV, but order of constituents may change according to the discourse context. For this reason, all six combinations of subject, object, and verb are possible in Turkish.

Turkish is head-final, meaning that modifiers always precede the modified item. For example, an adjective (modifier) precedes the head noun (modified item) in a noun phrase. In the basic word order of the sentence, the subject and the object (modifiers) precede the verb (modified item). Although the head-final property can be violated at major constituent levels (SOV) of a sentence, it is preserved at sub-clause levels and smaller syntactic structures. For example, the following simple noun phrase demonstrates this property.

- (the girl with the red hat)
- kırmızı şapkalı   kız
- red   with hat   girl

In this phrase, the adjective "kırmızı" modifies the noun "şapka", and the phrase "kırmızı şapkalı" modifies the noun "kız".

Like all other Altaic languages, Turkish is agglutinative. Non-functional words can take many derivational suffixes and each of these derivations can take its inflectional suffixes. In addition, in Turkish, inflectional suffixes have important grammatical roles. Inflectional suffixes of intermediate derived forms of a word also contribute to these syntactic roles of the word. Hence, there is a significant amount of interaction between syntax and morphotactics. For example, case, agreement, relativization of nouns and tense, modality, aspect, passivization, negation, causatives, and reflexives of verbs are marked by suffixes. For example, the following single Turkish word contains two derivational morphemes, and it corresponds to a complete English sentence.

- (you had not been able to make him do)
- yaptıramıyormuşsun
- yap+tır$_1$+amı$_2$+yor$_3$+muş$_4$+sun$_5$
- yap+Verb ^DB+Verb+Caus$_1$ ^DB+Verb+AbleNeg$_2$+Neg +Prog1$_3$ +Narr$_4$ +A2sg$_5$

In this example, "^DB" indicates the derivational morpheme boundary, and the underlined morphemes are derivational morphemes.

4. System Architecture

The system architecture of Turkish parser is depicted in Figure 1 as a flowchart by labeling the parsing steps 1 through 5. The parser uses the Turkish morphological analyzer and the link grammar static libraries externally. A given sentence is transformed into certain intermediate forms at each step, and at the end all possible linkages of the sentence are generated by the parser. In the rest of this section, each step is explained separately.

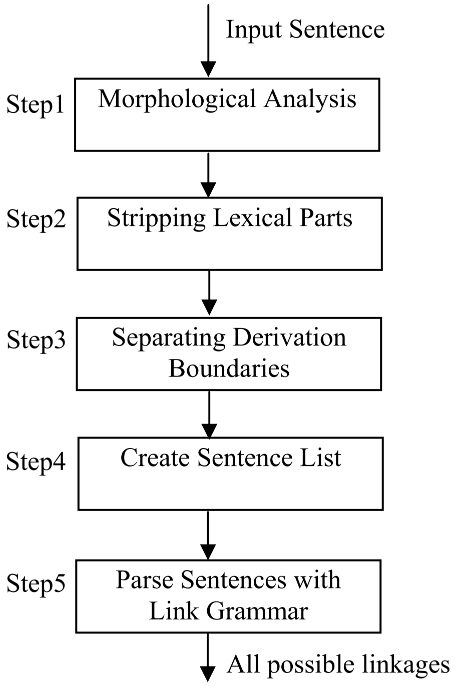

Figure 1. System Architecture of Turkish Parser

Step 1 - Morphological Analysis:

After taking the input sentence in step 1, the system calls the external morphological analyzer for each word of the sentence to get its morphological structure. A fully functional Turkish morphological analyzer is used in the analysis of the words. The word itself is used in the rest of the system if the morphological analyzer cannot analyze a word.

For example, if the following input sentence is given into step 1, the output from step 1 will be as follows.

```
Input to Step 1:

• sen kitabı okudun
• (you read the book)

Output from Step 1:

• sen  (you)
  i.sen+Pron+A2sg+Pnon+Nom

• kitap (book)
  i.kitap+Noun+A3sg+Pnon+Acc
  ii. kitap+Noun+A3sg+P3sg+Nom

• oku (read)
  i. oku+Verb+Pos+Past+A2sg
```

Step 2 - Stripping Lexical Parts:

In step 2, the output of step 1 is preprocessed for the following parsing stages. In this step, lexical parts of the words are removed for all types of words except conjunctions. In fact, Turkish link grammar is designed for the classes of word types and their feature structures, i.e. POS, rather than the words themselves.

When the above output from step 1 is given into step 2, the lexical parts are removed from the morphological structures of the words, and the following output is created in step 2.

```
Output of Step 2:

• sen  (you)
  i.Pron+A2sg+Pnon+Nom

• kitap (book)
  i.  Noun+A3sg+Pnon+Acc
  ii. Noun+A3sg+P3sg+Nom

• oku (read)
  ii. Verb+Pos+Past+A2sg
```

The output of step 2, as shown above, is the list of unlexicalized morphological feature structures of words.

Step 3 - Separating Derivation Boundaries:

If a word is derived from another word by the help of at least one derivational suffix, then its feature structure must contain at least one derivational boundary. Feature structures of words with derivational boundaries are handled in a special way in our system. In step 3, the words are separated at derivational boundaries and the part of speech tag of each derived form is marked in order to indicate its position in that word. The algorithm for step 3 is given in Figure 2. After step 3, a derived word is represented with a sequence of tokens. Each token starts with a part of speech tag with a position mark, and continues with inflectional feature structures. Below are some examples for step 3.

```
Input:
    Noun+A3sg+Pnon+Acc
Output:
    Noun+A3sg+Pnon+Acc

Input:
    Noun+A3sg+P1pl+Loc^DB+Adj+Rel
    ^DB+Noun+Zero+A3sg+Pnon+Gen
Output:
    NounRoot+A3sg+P1pl+Loc
    AdjDB
    NounDBEnd+A3sg+Pnon+Gen
```

Since the first example does not contain any derivation, no action is taken and the part of speech tag "Noun" at the

```
if    ( the feature structure of input word has
        no derivational boundary)
      • Output is equal to input
else {
      • Separate the word from the derivational boundaries to create a list of derived forms $DF_1$ ... $DF_n$
        where $n \geq 2$. In this list, $DF_1$ is the root word, $DF_n$ is the last derivation, and others are intermediate
        derivations.
      • Replace POS tag portion of $DF_1$ with the concatenation of POS of $DF_1$ and the string "Root".
      • Replace POS tag portion of $DF_n$ with the concatenation of POS of $DF_n$ and the string "DBEnd".
      • Replace POS tag portion of each intermediate derived form with the concatenation of POS of that
        intermediate form and the string "DB".
      • Output is the list of derived forms
   }
```

Figure 2. Separating Words from Derivation Boundaries

beginning of the output indicates that it is a noun without a derivation.

The second example above is divided into three derivational forms. In the example, the POS tag portion of each derived form is underlined, and they are replaced by new strings as described in the algorithm given in Figure 2 in order to indicate their positions in the word. After step 3, each token starts with a part of speech tag (or a part of speech tag followed by one of the strings “Root”, “DB”, or “DBEnd”) and continues with inflectional suffixes. The first token starts with “NounRoot”, and it indicates that the root word is a noun and that token is the root word of the derived word. “AdjDB” in the second token indicates that the word is converted into an adjective with a derivational morpheme, and that token is an intermediate derivation of the word. “NounDBEnd” in the last token indicates that the word is reconverted back into a noun again with a derivational morpheme, and that token is the last derivation of the word.

Step 4 - Create Sentence List:

Since a part-of-speech tagger is not used is our system, the number of feature structures found for the words is very large. For this reason, after step 4, a separate sentence is created for each of the morphological parse combinations of the words in step 3. For the example sentence given in step 2, “sen kitabı okudun” (you read the book), the output of step 4 is shown below.

```
Input to Step 4:

      i.  Pron+A2sg+Pnon+Nom

      i.  Noun+A3sg+Pnon+Acc
      ii. Noun+A3sg+P3sg+Nom

      i.  Verb+Pos+Past+A2sg

Output from Step 4:

      i.  Pron+A2sg+Pnon+Nom
          Noun+A3sg+Pnon+Acc
          Verb+Pos+Past+A2sg

      ii. Pron+A2sg+Pnon+Nom
          Noun+A3sg+P3sg+Nom
          Verb+Pos+Past+A2sg
```

This means that the sentence has two different representations at the morphological level. Each output is a sequence of tokens, and the first part of each token is a part of speech tag (or a part of speech tag with derivation position information). The rest of each token contains only the inflectional suffixes.

Since each word more than one representation at the morphological level, a sentence can have many representations at the morphological level. Each representation of the sentence will be fed into the parser at Step 5. In the future, the number of possible representations of the sentence at the morphological level will be reduced as a result of the integration of a Turkish morphological disambiguator into the system.

Step 5 - Parsing Sentences:

At the end, for each of these sentences, the link grammar is called, and each of the sentences is parsed in step 5 with respect to the designed Turkish link grammar. The Turkish link grammar contains a set of link requirements for each part of speech tag (or a part of speech tag followed by one of the strings “Root”, “DB”, or “DBEnd”).

A linking requirement is written for a token, and the link requirements of a token depend on the part of speech tag of the token, and the inflection suffixes in that token. Each link requirement may contain left and right linking requirements.

If all the linking requirements of the tokens in a sentence are satisfied, a linkage is created and returned as an output of the parser for the sentence. There is more than one possible linkage connection between tokens; all linkages are returned as the outputs of the parser.

5. Linking Requirements Related to Agglutination

In order to preserve the syntactic roles that the intermediate derived forms of a word play, they are treated as separate words in the grammar. On the other hand, to show that they are the intermediate derivations of the same word, all of them are linked with the special “DB” (derivational boundary) connector. In the following example, the feature structure of each morpheme is marked with the same subscript.

- uzman$_1$+laş$_2$ (specialize)
- uzman+Noun+A3sg+Pnon+Nom$_1$ ^DB+Verb+Pos+Imp+A2sg$_2$
- NounRoot+A3sg+Pnon+Nom$_1$ VerbDBEnd+Pos+Imp+A2sg$_2$

```
  +-----------DB-----------+
  |                        |
NounRoot+A3sg+Pnon+Nom VerbDBEnd+Pos+Imp+A2sg
```

Here, the noun root "uzman"(specialist) is an intermediate derived form and connected to the last derivation morpheme "-laş" (to become) by the "DB" link, to denote that they are parts of the same word. Since the root word (NounRoot) is an intermediate derivation form of this derived word, it can only have left linking requirements by contributing the left linking requirements of the derived word. The last derived form (VerbDBEnd) can have both left and right linking requirements. In general, a derived word consists of a sequence of intermediate derived forms where the first one is the root word, and the last derivation form. However, these intermediate derived forms, IDF, do not contribute to the right linking requirement of the last derived word. In addition, the "DB" linking requirements of the intermediate derived forms are different according to their order. The last derived form can contribute to both left and right linking requirements of the derived word.

In Figure 3, linking requirements of a word, with n intermediate derived forms (IDF$_1$...IDF$_n$) are illustrated. In Figure 3, "LL" represents the links to the words on the left hand side of the word, and "RL" represents the links to the words on the right hand side of the word. IDFs of the word are connected by "DB" links. As it can be seen all ***n*** IDFs can connect to the words to the left of, but only the last IDF, IDFn can connect to the words on the right hand side of the word. In addition, IDF$_1$, which is the root stem, needs only to connect to its right with the "DB" connector, whereas the last IDF (IDFn) needs to connect to its left with the same connector. On the other hand, all the IDFs between these two should connect to both to their lefts and to rights with "DB" links to denote that they belong to the same word. Hence, the same IDF, has different linking requirements depending on its place in a word. To handle this situation, different items are placed into the grammar representing each of these three places of the same word.

Tokens with a part of speech tag (without any derivational position marker) can have left and right linking requirements. We call these linking requirements as "non-derivational linking requirements" (NDLR). In addition, NDLLR is used as an abbreviation for "non derivational left linking requirement" and NDRLR is for "non derivational right linking requirement". Thus, all tokens with a part of speech tag without a derivational position marker will only have NDLR.

Tokens containing a part of speech tag with a derivational position marker may not use all NDLR, and they can have "DB" linking requirements. Their linking requirements depend on their position in the derived word. Figure 4 gives linking requirements of tokens with a part of speech tag with derivational position marker, and they are referred as IDFs (intermediate derivational forms) in Figure 4. In Figure 4, derivational linking requirements are in italics and non-derivational linking requirements are in bold.

As it can be seen in Figure 4, NDLRs of an IDF placed at the beginning and in the middle are the same. In addition, NDLR of the IDF for these two positions is a subset of the whole NDRL of the same IDF placed at the end.

6. Performance Evaluation

The performance of our system is tested for coverage with a document consisting of sentences collected from domestic, foreign, sports, astrology, and finance news randomly together with sentences from a storybook for children. Before beginning testing, punctuation symbols are removed from the sentences. In addition, incorrect morphological analyses are removed from the results. Table 1 shows the

```
-----------------LLn---------+
--------------LLn-1----+     |
---------LL2----+      |     |
--LL1--+        |      |     |
       +---DB---+- ...-+-DB--+-RL--
       |        |      |     |
   IDF1(Root)  IDF2 .. IDFn-1 IDFn
```

Figure 3. Linking Requirements of Intermediate Forms of a Word

// linking requirements of the "intermediate
// derived form at the beginning", IDF$_{Root}$
IDF$_{Root}$: **NDLLR** & *DB+*;

// linking requirements of the same "intermediate
// derived form in the middle", IDF$_{DB}$
IDF$_{DB}$: *DB-* & **NDLLR** & *DB+*;

// linking requirements of the same "intermediate
// derived form at the end", IDF$_{DBEnd}$
IDF$_{DBEnd}$: *DB-* & **NDLLR** & **NDRLR**;

Figure 4. Linking Requirements of an IDF According to Its Place

Table 1. Statistical Results of the Test Run

Number of Sentences	250
Average number of words in each sentence	5.19
Percentage of the sentences for which resulting parses contains the correct parse	84.31
Average number of parses	7.49
Average ordering of the correct parse	1.78

results of the test run.

In the experiment, 250 sentences are used. Average number of words in the sentences is 5.19. Average number of parses per sentences is 7.49. However, for two of the sentences, the number of the parses are very high, i.e. 22 and 50. Both of these two sentences contain many consecutive nouns. Since nouns are not subcategorized for time, place, and title, this resulted in many incorrect indefinite and adjectival nominal groups to be generated and this is the problem in these two sentences. Moreover, one of these sentences consists of words with very complex derivational morphotactics, i.e. many derivational intermediate forms, which results in the number of possible links between these intermediate derived forms to increase. In addition, for 84.31% of the sentences, the result set of the parser contains the correct parse. Lastly, average ordering of the correct parse in the result set was 1.78. However, for 62.39% of the sentences, the first parse is the correct parse and for 80.94% of the sentences, one of the first three parses is correct.

7. Conclusions and Future Work

In this work, we have developed a grammar of Turkish language in the link grammar formalism. Noun phrases; postpositional phrases; dependent clauses constructed by gerunds, participles, and infinitives; simple, complex, conditional, and ordered/compound sentences; nominal and verbal sentences; regular sentences; positive, negative, imperative, and interrogative sentences; pronoun drop; freely changing order of adverbial phrases, noun phrases acting as objects, and subject are in the scope. In addition, quotations, numbers, abbreviations, hyphenated expressions, and unknown words are handled. However, inverted sentences, idiomatic and multi-word expressions, punctuation symbols, and embedded and some types of substantival sentences are currently out of the scope.

In the grammar, we used a fully described morphological analyzer, which is very important for agglutinative languages like Turkish. The Turkish link grammar that we developed is not a lexical grammar. Although we used the lexemes of some function words, we used the morphological feature structures for the rest of the word classes. In addition, we preserved the syntactic roles of the intermediate derived forms of words in our system by separating the derived words from their derivational boundaries and treating each intermediate form as a distinct word.

As mentioned above, because of the productive morphology of Turkish, our linking requirements are defined for morphological categories. However, instead of using only the morphological feature structures of words, stems of words can also be added to the current system. Thus, the results of our current Turkish link grammar can be more precise. In addition, statistical information about the relations between the words can be embedded into the system. Moreover, our current system does not use a POS tagger, and its addition will improve the performance of the system in terms of both time and precision. During the tests, we recognized that there are many multi-word expressions in Turkish and a multi-word expression processor is necessary.

Although the adopted unique link grammar approach is used in the development of a Turkish link grammar, it can be used in the development of the link grammars for other languages with complex morphology. The adopted approach can provide flexibility in the development of link grammars for such languages.

8. References

[1] Bozşahin, C. and Göçmen, E. 1995. *A Categorial Framework for Composition in Multiple Linguistic Domains*. In Proceedings of the Fourth International Conference on Cognitive Science of NLP, Dublin, Ireland.

[2] Demir, Coşkun. 1993. *An ATN Grammar for Turkish*. M.S. Thesis, Bilkent University.

[3] Eryiğit, G., and Oflazer, K. 2006. *Statistical Dependency Parsing of Turkish*. In Proceedings of EACL 2006 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy.

[4] Güngördü, Zelal. 1993. *A Lexical Functional Grammar for Turkish*. M.S. Thesis, Bilkent University.

[5] Hoffman, Beryl. 1995. *The Computational Analysis of the Syntax and Interpretation of 'Free' Word Order in Turkish*. PhD thesis, University of Pennsylvania.

[6] Oflazer, K. 1999. *Dependency Parsing with an Extended Finite State Approach*. In Proceedings of 37th Annual Meeting of the ACL, Maryland, USA.

[7] Oflazer ,K.; Say ,B.; Hakkani-Tür, D.K.; Tür, G. *Building a Turkish Treebank. Invited chapter in Building and Exploiting Syntactically-annotated Corpora*, Anne Abeille Editor, Kluwer Academic Publishers. The treebank is available online at: http://www.ii.metu.edu.tr/~corpus/treebank.html

[8] Sleator, D. D. K. and Temperley, D. 1993. *Parsing English with a Link Grammar.* Third International Workshop on Parsing Technologies.

[9] Şehitoğlu, O. Tolga. 1996. *A Sign-Based Phrase Structure Grammar for Turkish*. M.S. Thesis, Middle East Technical University.