# A COMPUTATIONAL INTERFACE FOR SYNTAX AND MORPHEMIC LEXICONS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY



BY

RUKET ÇAKICI



IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

SEPTEMBER 2002

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Tayfur Öztürk
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Cem Bozşahin
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Ferda Nur Alpaslan

Dr. Ayşenur Birtürk

Assoc. Prof. Dr. Cem Bozşahin

Assoc. Prof. Dr. Bilge Say

Prof. Dr. Deniz Zeyrek

# ABSTRACT

A COMPUTATIONAL INTERFACE FOR SYNTAX AND MORPHEMIC LEXICONS

Çakıcı, Ruket

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Cem Bozşahin

September, 66 pages

The role and the place of the morphology component in language faculty is still a point of debate in linguistics. The bracketing mismatches in syntax and morphology and the phrasal scope of morphemes provide evidence for the existence of a problem for morphology and semantics. In fact, the mismatches in syntax and morphology have semantic basis, which is largely ignored in studies of inflectional morphology. A possible solution to this problem is proposed by Bozşahin (2002a), which provides a purely morphemic lexicon and a CCG parser, that constructs syntactic constituents in parallel to semantic interpretations. However, the lack of a mechanism to handle morphotactics and morphophonemics not only puts a burden on lexicon by having all allomorphs of the same morpheme as separate lexical entries, but also prevents the ambiguity which must be perserved until interpretation. In this study, a morphological analyzer which provides the morphemic parser with a stream of morphemes together with their morphosyntactic and semantic categories is presented. The objective is to compare the three different architectures of lexicon-morphology-syntax interface (Bozşahin,

2002a). As a consequence, a modular system is proposed, and the coverage of the combinatory morphemic lexicon on transparent correspondence of morphosyntax and semantics is maintained.

Keywords: NLP, morphosyntax, morphology, lexicon, CCG, computational linguistics

# ÖZ

SÖZDİZİM VE BİÇİMBİRİMSEL SÖZLÜKLER İÇİN BERİMSEL ARAYÜZ

Çakıcı, Ruket

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Assoc. Prof. Dr. Cem Bozşahin

Eylül, 66 sayfa

Biçimbilim bileşeninin dil yapısı içindeki yeri ve görevi dilbilimde halen tartışma konusudur. Sözdizim ve biçimbilimdeki parantezleme çelişkileri ve çekim eklerinin öbeksel kapsamı olarak bilinen olgular biçimbilim ve semantik arasında bir problem olduğunun göstergesidir. Sözdizim ve biçimbilim arasındaki bu çelişkilerin kökeni anlambilimdir fakat bu, biçimbilim çalışmalarında çoğunlukla gözardı edilir. Bozşahin (2002a) buna çözüm olarak CCG çözümleyici ile birlikte kullanılacak olan biçimbirimsel bir sözlük önermiştir. Bu çözümleyici paralel bir şekilde sözdizimsel ve semantik yapıyı oluşturur. Fakat biçimdizimsel ve fonolojik yapıların analizi için bir sistem olmaması, yalnızca yükü sözlüğün üzerine yıkmakla kalmayıp çözümlemeye kadar korunması gereken belirsizliği de engeller. Bu çalışmada bir biçimbilim çözümleyicisi, morfosentaktik ve semantik bilgileri ile birlikte biçimbirim listesini çözümleyiciye gönderir. Amaç, Bozşahin'de (2002a) verilen 3 değişik mimariyi sözlük, biçimbilim ve sözdizim arasındaki ilişki açısından karşılaştırmaktır. Sonuç olarak modüler bir sistem elde edilirken, sözdizim ve semantiğin şeffaflığı ilkesi de korunmuş olur.

Anahtar Kelimeler: Doğal dil işleme, morfosentax, biçimbilim, sözlük, CCG, berimsel dilbilim

To my family

# ACKNOWLEDGMENTS

I would like to thank my advisor Cem Bozşahin for his invaluable guidance and support in conducting this research. The knowledge and vision provided by him led to overcome many problems that I faced throughout this thesis.

I am grateful to all of my friends and to the staff in the faculty for their accompany and help in the past three years. My special thanks are to Semra Doğandağ, Ulaş Yılmaz and Ersan Topaloğlu for their invaluable help and for being such wonderful friends.

I would like to thank my jury members; Deniz Zeyrek, Bilge Say, Ayşenur Birtürk, Ferda Nur Alpaslan and also Onur Tolga Şehitoğlu and Erkan Korkmaz for their helpful comments and suggestions.

Last but not least, I would like to thank my family and Levent Tavşancı for their endless support and patience.

# TABLE OF CONTENTS

ABSTRACT . . . . . iii

ÖZ . . . . . v

DEDICATON . . . . . vii

ACKNOWLEDGMENTS . . . . . viii

TABLE OF CONTENTS . . . . . ix

LIST OF TABLES . . . . . xi

LIST OF FIGURES . . . . . xii

LIST OF ABBREVIATIONS . . . . . xiii

CHAPTER

1 INTRODUCTION . . . . . 1

2 BACKGROUND . . . . . 10

2.1 Turkish . . . . . 10

2.1.1 Morphology . . . . . 12

2.1.1.1 Nominal Paradigm . . . . . 12

2.1.1.2 Verbal Paradigm . . . . . 13

2.1.2 Syntax . . . . . 17

2.1.3 Word Order . . . . . 18

2.2 PC-KIMMO and Two-level Morphology . . . . . 19

2.2.1 The Lexicon . . . . . 20

2.2.2 The Phonological Component . . . . . 21

2.2.3 Word Grammar . . . . . 22

2.2.4 Computational Properties . . . . . 23

2.3 Combinatory Categorial Grammars . . . . . 23
2.3.1 CG Formalism . . . . . 24
2.3.2 Combinatory Categorial Grammars (CCG) . . 25
2.3.3 Computational Power of CCGs . . . . . 26
3 ARCHITECTURE OF THE MORPHEMIC LEXICON . . . . . 28
3.1 Morphological Parser . . . . . 32
3.2 The Phonological Rules . . . . . 33
3.3 The Lexicon . . . . . 34
3.4 The Word Grammar . . . . . 37
3.4.1 Nominal Paradigm . . . . . 37
3.4.1.1 -ki relativization . . . . . 38
3.4.1.2 Adjectives . . . . . 39
3.4.1.3 Pronouns . . . . . 39
3.4.1.4 Compound Nouns . . . . . 40
3.4.1.5 Predicative . . . . . 41
3.4.2 The Verbal Paradigm . . . . . 42
3.5 The Lexicon-Mophology-Parser Interface . . . . . 42
4 EXPERIMENTAL RESULTS . . . . . 49
5 CONCLUSION . . . . . 53
APPENDICES . . . . . 54
A How to create lexical entries for the system . . . . . 55
REFERENCES . . . . . 60

# LIST OF TABLES

4.1 The comparison of parsing performances of the current system with Bozşahin's(2002a) Combinatory Morphemic Lexicon (CML). We assume normal form parsing in all cases. . . . . 51

A.1 The diacritics for Turkish . . . . . 59

# LIST OF FIGURES


1.1 Architectures for morphology-Lexicon-Syntax Interface (from Bozşahin (2002a)) . . . . . 8

2.1 The Nominal Paradigm . . . . . 12
2.2 The Verbal Paradigm . . . . . 14
2.3 Rule types for two-level morphology . . . . . 21

3.1 Architecture . . . . . 28
3.2 Feature structure representation for accusative case morpheme . 46
3.3 A sample of autogenerated sublexicon entries . . . . . 47

A.1 The word grammar definition of the category for plural morpheme 56
A.2 An example of a complex morphosyntactic category . . . . . 57
A.3 The word grammar definition of the category for pronouns . . . 58

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ACC | Accusative case suffix |
| ABIL | Abilitative (modality) suffix |
| ADJ | Derivational suffix making an adjective from a noun |
| ADV | Adverb |
| AGR | Agreement suffix |
| AOR | Aorist suffix |
| AOR0 | Aorist 0 morpheme |
| CAUS | Causativization suffix |
| CM | Compound marker |
| COMPOUNDN | Compound Noun |
| COPULA | Copula suffix |
| DAT | Dative case suffix |
| GEN | Genitive suffix |
| gen.pl | Genitive plural |
| gen.sg | Genitive singular |
| INF | Infinitive suffix |
| LOC | Locative case suffix |
| N | Noun |
| NEG | Nagative suffix |
| nom.sg | Nominative singular |
| nom.pl | Nominative plural |
| OBLIG | Obligatory (modality) suffix |
| PASS | Passivization suffix |
| PAST | Past tense suffix |
| PLU | Plural suffix |
| POSS | Possessive suffix |
| POSSIB | Possibility (modality) suffix |
| POSS2PL | Possessive second person plural suffix |
| POSS3PL | Possessive third person singular suffix |
| POSS3SG | Possessive third person singular suffix |
| PERS2PL | Second person singular suffix |
| PERS3PL | Third person plural suffix |
| PERS1SG | First person singular suffix |
| PERS2SG | Second person singular suffix |
| REL | *-ki* relativizer suffix |
| TENSE | Tense suffix |

# CHAPTER 1

# INTRODUCTION

There are several ways to conceive the morphology-lexicon-grammar relation in a computational system for language processing. Traditionally, morphology deals with internal structure of words, lexicon is repository of words as unanalysed integral units, and grammar pays no attention to the units smaller than words. [1]

A typical NLP design for the morphology-lexicon-grammar interaction is having a separate morphological component whose database of words contains minimal or no syntactic-semantic information. This component breaks the words into their stems and affixes, so that subsequently the system can perform lexical access using the stem, and the resulting lexical choice is fed to the grammar component. This design is common in NLP systems for morphologically simple languages such as English. The basic assumption in this design is that only words and larger units (phrases) contribute to meaning composition in grammar so that only words are kept in the lexicon, and no structural information about smaller units are stored in morphological database, lexicon or grammar. But this assumption can be challenged for morphologically simple languages as well

[1] This view dates back to the beginning of generative grammar in 1960s (Chomsky, 1959). Chomsky's (1995) latest version of generative grammar, called the minimalist program (MP), still adheres to this principle: a numeration in MP is the only thing that the computational system (i.e., the grammar) sees as part of its syntactic processing. A numeration in MP is the choice of a set of fully inflected words from the lexicon.

as morphologically complex languages.

Pesetsky (1979) observed that the morphophonemic restrictions while attaching a morpheme may be different from the semantic bracketing as the following example from English shows.

(1) a. [*un*-[*happy*-*er*]]

b. [[*un*-*happy*]-*er*]

In English, an adjective may take comparative form only if it is monosyllabic, throchaic or disyllabic. For example *happy* or *easy*. So *happy* can take *-er* but *unhappy* cannot, normally. The bracketing should be as in (1a) according to the morphophenemic restrictions. However, the actual meaning of the word is '*more unhappy*' rather than '*not happier*'. This means the semantic interpretation is the bracketing in (1b). This is a mismatch between morphology and semantics.[2]

Another type of these bracketing paradoxes is cited in (Spencer, 1991, p.398). Examples in (2) are taken from Williams (1981). In this type of paradox, the morphological attachment characteristic of the affix contradicts with the semantic scope it covers. This causes a mismatch between the morphological and semantic bracketing. The bracketings in (2a-d) correspond to morphological bracketings and (2e-h) correspond to the semantic bracketings. For example the *-ian* suffix in (2d) applies to the entire phrase to have the interpretation in (2h): '*a linguist who studies on transformational grammars,*' not the one implied by the morphological bracketing when the suffix attaches to the second word in the phrase, which leads to the meaning: '*a grammarian who is transformational.*'

(2) a. [*Godel* [*number*-*ing*]] e. [[*Godel number*]-*ing*]

b. [*hydro*-[*electric*-*ity*]] f. [[*hydro*-*electric*]-*ity*]

c. [*atomic* [*scient*-*ist*]] g. [[*atomic science*]-*ist*]

d. [*transformational* [*grammar*-*ian*]] h. [[*transformational grammar*]-*ian*]

[2] Some authors such as Stump (1991) claim that this is not a mismatch.

This problem is not limited to English, which is a morphologically simple language. Pesetsky gives a variety of these kinds of examples from Russian (Pesetsky, 1979; Pesetsky, 1985). The following German example is taken from (Muller, 1999, p.401). Semantics given in (3b) is in conflict with the (3a) since the semantic scope of *habt* differs from the its attachment characteristics.

(3) a. *Wenn* [*Ihr Lust*] *und* [*noch nichts anderes vor-*]*habt,*
if you pleasure and yet nothing else intend

*können wir sie ja vom Flughafen abholen*
can we them PARTICLE from.the airport pick up
'If you feel like it and have nothing else planned, we can pick them up at the airport.'

b. *Ihr Lust habt UND noch nichts anderes vorhabt*

Sadock (1991) gives examples from Greenlandic, an ergative language.[3] In Sadock's example, 'appear, seem' morphologically attaches to 'love' but semantically it has scope over the entire phrase.

(4) Kaali-p Amaalia asa-gunar-paa
Karl-erg Amaalia(abs) love-appear-indic/3sg
*'Karl seems to love Amaalia.'*

Bozşahin (2002a) provides examples of phrasal scope of inflections from Turkish morphology. For instance the possessive and dative affixes in (5) scope over the entire phrase of '*eş ve çocuk*' so that the semantic bracketing is as shown.

(5) adam-ın [eş ve çocuğ]-u-na
Man-GEN spouse and child-POSS-DAT
*'(to) the man's spouse and child'*

[3] "A term used to express the formal parallel between the object of a transitive verb and the subject of an intransitive one (i.e. they display the same case). The subject of the transitive verb is referred to as 'ergative' whereas the subject of the intransitive verb, along with the object of the transitive verb, are referred to as 'absolutive'." (Crystal, 1998, p.138)

Phrasal scope of inflectional suffixes are also seen in other syntactic processes such as subordination:

(6) a. Ayşe kalem-i isti-yor.
Ayşe kalem-ACC want-TENSE
*'Ayşe wants the pencil.'*

b. Ayşe [Ahmet'in kitab-ı oku-ma-sı]-nı isti-yor.
Ayşe Ahmet-GEN book-ACC read-INF-AGR-ACC want-TENSE
*'Ayşe wants Ahmet to read the book.'*

As (6a) shows *'iste (want)'* verb expects an accusative marked object and this applies to cases like (6b) in which the complement is a nominalized predicate. The case of the nominalized verb ranges over the entire embedded clause.

These mismatches brings out the question: Is morphology a separate module or is spread over syntax and semantics? There are several views about this issue.

Sadock (1991) provides examples to bracketing paradoxes such as (4). In his theory, *Autolexical Syntax,* he provides a model in which morphology is a separate module of the grammar, used in determining grammatical well-formedness, where morphological structure may be nonisomorphic to syntactic or semantic structure. This nonisomorphism is constrained by some principles, namely *Incorporation Principles.*

Sproat (1985; 1998) however suggests that there is no separate module of morphology but the representation of the words is distributed over different components. He proposes two structures for a word. These are (word-) syntactic and (word-) phonological structures which he relates with surface structure (S-structure) and phonological form (PF) respectively. These structures may be nonisomorphic. But phonological structure is mapped to syntactic structure by his *Mapping Principle.* All affixes are given a phonological representation and a morphosyntactic representation. Phonological attachment is constrained by the mapping principle while the syntactic structure is being constructed in order to maintain isomorphy. Then the linear ordering of the morphemes are retained

by a process led by some operators. For example (7a) is mapped to (7b).

(7) a. [un[easy er]]

b. [[un easy] er]

Sproat claims that if sentences have representations that are distributed on S-structure and PF, then words may also have the same property, and moreover, that syntax of words is properly a part of syntax.

There has been several attempts to make a sound representation of the lexicon. Beard (1995), Aronoff (1976), and Matthews (1972) proposed that only content morphemes have separate lexical entries and all function morphemes, including affixes are represented by word-formation rules (Sproat, 1998). Sproat (1985), following Lieber(1980), assumes a purely morphemic lexicon.[4] Bozşahin (2002a), which is followed in the current study, claims that a lexemic lexicon in which only content words are represented is not suitable for handling the interface problems between morphology and syntax. What he proposes is a morphemic lexicon in which all inflectional morphemes as well as words are represented as structural units.

There are various computational studies dealing with interface problem in Turkish. These provide a spectrum of modularity and integration between linguistic components. Güngördü and Oflazer (1995) provide a framework in which the morphology and syntax are separate components and syntactic parsing is done after morphological parsing. The system assumes all derivational and inflectional morphology is handled in morphology module. And the result is passed to an LFG parser. In (Şehitoğlu and Bozşahin, 1999) derivations are listed in the lexicon while the inflectional morphology is handled with lexical rules of HPSG. This will also not solve the bracketing problems but only pass them to the other levels of linguistic representation for considering rebracketing. Hoffman (1995) provides a categorial approach which is truly lexical. All inflected

[4] Here by morphemic we mean that all content words, affixes and clitics have separate entries.

and derived forms are listed in the lexicon which leaves no possibility for handling phrasal scope of inflections. Bozşahin and Göçmen (1995) is an attempt to provide a framework in which morphological, syntactic and semantic compositions are done in parallel. These domains are treated as uniform rather than different domains. And word and phrase structure are not distinguished. However, overgeneration results, because of inability of the system to make finer distinctions in morphosyntactic types. The system in fact lacks the notion of types. And this causes a nontransparent integration of morphology and syntax as noted by Bozşahin (2002a). Bozşahin (2002a) provides a framework which has flexible constituency and flexible attachment properties. The structural units are no longer words but morphemes which can specify their attachment characteristics and semantic domains. This provides a transparent interface of morphology, syntax and semantics, and brings modularity by handling the interface problems.

In this formalism each morpheme has a phonological representation, a syntactic category and a semantic interpretation. The ordering constraints of the morphemes are handled by a lattice structure. Lattice structure has the primary role of regulating the transparent scoping in syntax and semantics. Combinatory Categorial Grammar (CCG) is the basis for the parser, and morphosyntactic modalities are used to constrain the categories.

Three potential NLP structures shown in Figure 1.1 are discussed in Bozşahin (2002a). The one that has a lexemic representation of lexicon and a separate morphological component is unsuitable because of the reasons that we discussed on previous pages. The proposal which is actually implemented in Bozşahin (2002a) is a morphemic lexicon which sends information to a parser where all the morphotactics, syntactic parsing and interpretation are handled. The morphemic lexicon provides the parser with morphemes together with their all morphosyntactic, and semantic categories. Having only a lexicon and a parser module, this model claims that inflectional morphology is actually a part of syntax. Bozşahin proposes another alternative in which input is fed to a mor-

phological analyser to be broken down to a stream of morphemes which will be matched with their semantics to be interpretable. The key distinction in this proposal with respect to other systems with morphological analysers is that the morphological analysis passes on semantic information to syntax as well, so that rebracketing due to semantics is not necessary. This has been the major problem in systems that isolates morphological information from syntactic and semantic structure. A matcher module —which we will show is not necessary— is included in this proposal to match the morphemes with their categories. This model may be preferred over the one implemented in (Bozşahin, 2002a) because of its generality, since morphological parsing liberates subsequent processing from language-specific morphological concerns and allows syntax and semantics to be specified transparently. Morphologically complex languages and morphologically simple languages are treated on an equal footing in terms of lexical category assignments for words and larger or smaller units. The lexicon of a inflectional language such as Russian need not worry about how to identify the stem and affixes, but concentrate only on their syntactic and semantic domains. The syntax of Turkish can be specified just as easily because it is relatively easier in Turkish to identify the morphs in the input. In fact this property of Turkish was exploited in Bozşahin (2002a) to have lexical type assignments to *surface* forms, e.g., assigning a category for *-da, -de, -ta, -te*, the locative case suffix. Using a similar strategy for Russian would require that all locative forms of nouns be listed in the lexicon, which misses the syntactic-semantic regularity of the locative case. A practical advantage of the current proposal is that the user need not know the notion of a morpheme; the input is taken as a list of words and broken into its segments by the morphological analyser. A further advantage of this aspect is that the ambiguity in the input is preserved. Since the user is not asked to segment the input, e.g., *kitapları* can be segmented as *kitap-POSS2PL, kitap-PLU-POSS3SG* or *kitap-PLU-ACC*. The disadvantage of Bozşahin's alternative proposal (Figure1.1(b)) is that there are several problems of modularity caused by matching process since some syntactic information needs

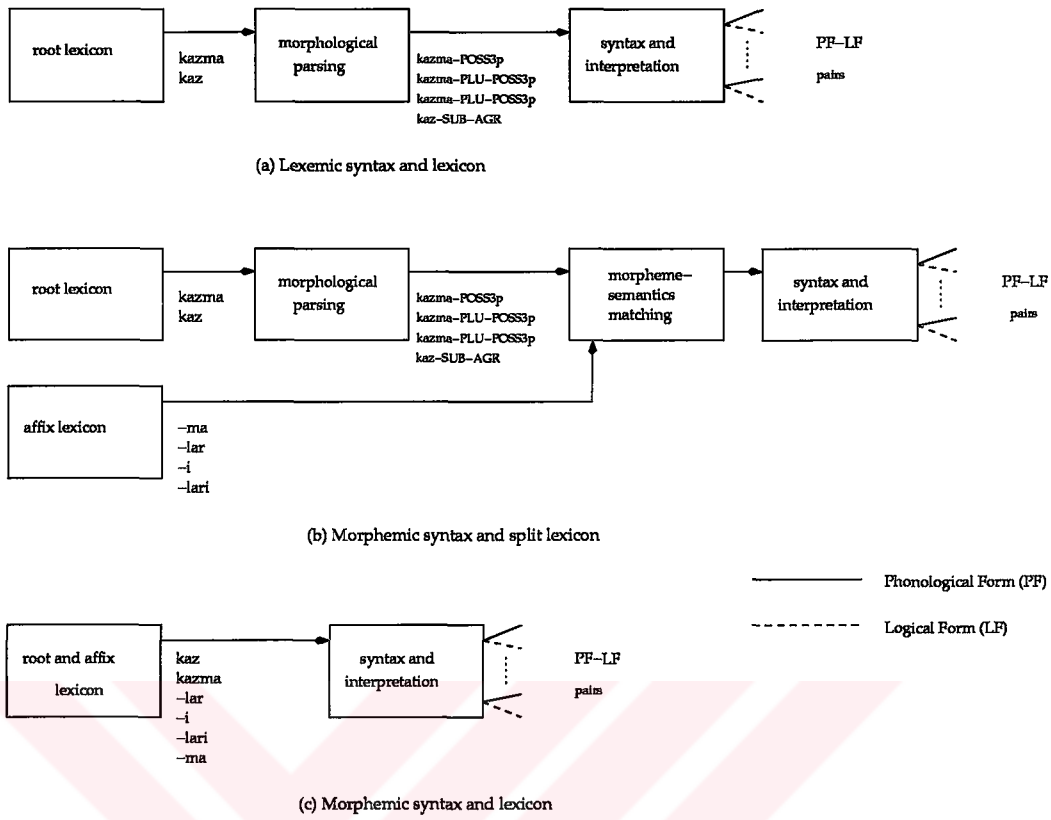


Figure 1.1: Architectures for morphology-Lexicon-Syntax Interface (from Bozşahin (2002a))

to be consulted in morphological parsing in cases when composition of affixes is necessary.

Our proposal is a morphemic lexicon together with a morphological analyzer and interpreter without the need for a matcher. Morphological analyzer will pass the morphosyntactic and semantic information of morphemes to the parser where they will be interpreted. This part is not different from the approach we explained. However the matcher will not be needed because the morphemes will carry all the possible category information in a list of categories when they go to the parser.

The morphological component will only concentrate on morphotactical constraints and will not deal with any syntactic or semantic constraint, which would cause a contradiction with modularity. It is based on two-level morphology (Koskenniemi, 1983) and is implemented in PC-KIMMO2 of Summer

Institute of Linguistics (Antworth, 1995). The working principles and the implementation is explained in Chapter 3.

In the rest of the thesis, we elaborate on these issues mentioned above. In the second chapter some background information about Turkish, the two level morphological model formalism, PC-KIMMO and Categorial Grammar is given. In the third chapter the basics of morphemic lexicon is explored and the architecture of the current proposal is discussed. Fourth chapter contains the results of the experiments and some computational comparisons between the new proposal and existing systems.

# CHAPTER 2

# BACKGROUND

## 2.1 Turkish

Turkish is a typical example of an agglutinating language, which means a language in which the words contain a linear sequencing of morphs. There are also isolating or inflecting languages in which words do contain more than one morpheme but unlike agglutinating languages, there is no one-to-one correspondence between these morphemes and linear sequence of morphs (Crystal, 1998). In agglutinating languages, grammatical elements are joined together in such a way that segmentation with the help of grammatical functions is relatively easy (Matthews, 1997; Underhill, 1986).

Turkish morphology is quite rich. This richness might make it difficult to see the different modules of syntax, semantics and morphology. This is because most of the time morphological constructs contain crucial syntactic information, or as we have seen in the introduction, bracketing paradoxes of morphological and semantic constituency may occur, which seems like there is a strong interaction between these domains. Some examples to these constructs are mentioned in Bozşahin (2002a) and Bozşahin and Göçmen (1995).

(8) a. uzun kol-lu gömlek
long sleeve-ADJ shirt

b. 1. [[uzun kol]-lu gömlek] *'a shirt with long sleeves'*

2. [[uzun] [kol-lu] gömlek] *'a long shirt with sleeves'*

(8) is an example of an ambiguity between two semantic interpretations caused by two different scoping domains of *-lu* suffix depending on attachment characteristics. Phrasal scope of the suffix leads to an interpretation as in (8a). while word scope causes an interpretation as in (8b). A complete NLP system must be able to construct both of these interpretations since these two can only be discriminated either by the use of discourse or prosody. This example reminds one of the famous telescope example in English, usually given as an example to the PP-attachment problem.

(9) *The lady saw the man with the telescope.*

Is it the man who has a telescope or the lady? This ambiguity is caused by the two different categories of *with* which modifies both nouns and verbs. The situation is the same for *-lu* suffix in Turkish which can have both word scope or phrasal scope. The actual meaning in both cases (in English and in Turkish) is determined by prosodic structure. This similarity cannot be just a coincidence and this makes one think that if *with* has a syntactic category in English *-lu* must also have one although it is a suffix.

These kind of ambiguities and bracketing paradoxes are not limited to derivational morphology. Consider the examples in (10). In (10a) the problem of constituency for dative case suffix is depicted. Plural and possessive suffixes apply to whole conjunct while they are only attached to the last word of the conjunct. The second sentence, (10b) is an example of a phrasal scope of inflection. The accusative case which the main predicate imposes on its argument is on the subordinate clause's verb but applies to whole clause.

(10) a. ev-in [kapı ve pencere] -ler -i

house-GEN door and window -PLU -POSS3SG

*'doors and windows of the house'*

b. Ahmet [adam-ın kitab-ı bana ver-me-si] -ni iste-di.
Ahmet man-GEN book-ACC me give-INF-AGR -ACC want-PAST
*'Ahmet wanted the man to give the book to me.'*

These kinds of constructions led to the theories of morphosyntax, in contrast to separation of syntax and inflectional morphology. Bozşahin (2002a) defines morphosyntax as "those aspects of morphology and syntax that collectively contribute to grammatical meaning composition." Assigning morphosyntactic categories and separate semantic interpretations to all morphemes will prevent the need for the rebracketing in subsequent processing. An NLP model should be proposed where semantic structure will be constructed in parallel to other levels of representation like morphological attachment, and phonological properties.

### 2.1.1 Morphology

Turkish has a complex inflectional and derivational system. The following are very simple representations of the nominal and verbal paradigms which constitutes the morphological system in Turkish. A more comprehensive analysis of Turkish morphology will be given in Section 3.4. Phonological system of Turkish is out of the scope of this thesis and will be omitted. For a detailed description of Turkish phonology the reader is referred to Kornfilt (1997), Pembeci (1998), and Öztaner (1996).

#### 2.1.1.1 Nominal Paradigm

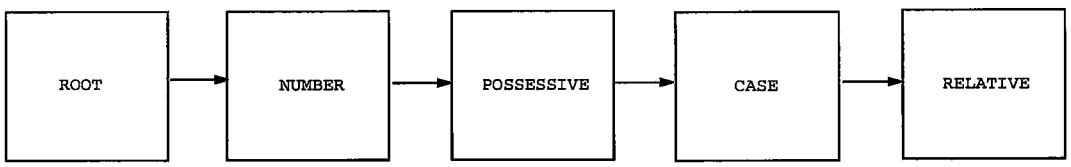


Figure 2.1: The Nominal Paradigm

The nominal paradigm (Figure 2.1) applies to derived or root nouns, adjectives and nominalized verbs. Nominal paradigm in Turkish is recursive by

means of a suffix, namely '-ki relativizer' which after affixes brings the stem to uninflected state so that all inflections can be aplied again.

(11) ev-de-ki-ler-in-ki
house-LOC-REL-PLU-GEN-REL
*'the ones that belong to the ones at home'*

If the case suffix just before the -ki suffix is a locative suffix, *-ki* acts like a derivational suffix and makes an adjective out of noun stem.

Nominal categories may also take predicative suffixes and become predicates.

(12) a. Sen [oyuncağı kırılan çocuk]-sun.
*'You are the kid whose toy broke.'*

b. Kırmızı olan [benim]-di.
*'The one which is red was mine.'*

c. Sen [benim annem]-sin.
*'You are my mother.'*

The brackets indicate the semantic scope of the inflectional suffixes.

#### 2.1.1.2 Verbal Paradigm

The verbal paradigm is richer than the nominal paradigm. A simple representation of verbal morphology in Turkish is given in Figure 2.2. Tense, modality, person and number agreement, polarity and subordination are all represented as inflectional markers on verbs in Turkish. An inflected verb on its own may constitute a sentence.

(13) Gel-e-me-yebil-ir-im.
Come-ABIL-NEG-POSSIB-TENSE-PERS1SG
*'I may not be able to come.'*

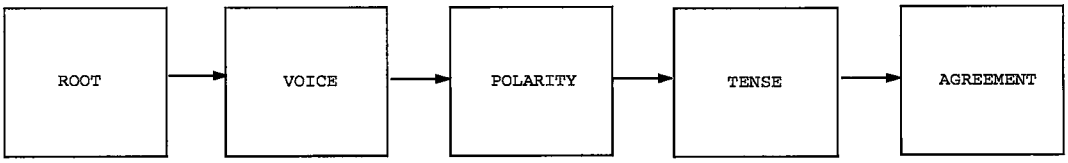


Figure 2.2: The Verbal Paradigm

Some inflections may change the voice and argument structure of the verbs like passivization and causativization. Passive construction decreases the valency of a verb while causative construction does the reverse (Kornfilt, 1997) as seen in (14a-d).

(14) a. Ahmet kitab-ı oku-du.
Ahmet book-ACC read-PAST
*'Ahmet read the book.'*

b. Ben Ahmet-'e kitabı oku-t-tu-m.
I Ahmet-DAT book-ACC read-CAUS-PAST-PERS1SG
*'I made Ahmet read the book.'*

c. Ahmet kitab-ı oku-du.
Ahmet book-ACC read-PAST
*'Ahmet read the book.'*

d. Kitap oku-n-du.
Book read-PASS-PAST
*'The book was read.'*

This causes another conflict with modularity principle: A morphological process changing the argument structure of a verb causes a nontransparency between morphology and syntax. If this change of argument structure is confined to a morphological component, how would the grammar know how much of syntactic-semantic structure is handled by the morphological component? If problems arise in interpretation, this is the question the parser faces. Its alternatives are rebracketing, or failure to provide an interpretation.

All the inflectional categories seen in Figures 2.1 and 2.2 are optional except for a few dependencies, e.g. a locative or genitive suffix before relativizer *-ki* in nominal paradigm or a tense marker before a person marker in verbal paradigm is obligatory. This means that a finite state system has to have a lot of states to handle all the possibilities that is caused by the absence of each of these categories. Oflazer's (1994) morphological analyser which uses this mechanism for handling the optional elements in morphotactics has 50 states for inflectional paradigm which expands to 30,000 states and 100,000 transitions when compiled together with morphophonemic rules.

Turkish also has a rich derivational morphology. The reason why derivational morphology is not explored is an interesting and curious point. But it can be one or all of the following reasons:

1. Derivational morphology is not considered compositional and productive. The same derivational suffix may have distinct semantic interpretations which are not predictable from the meaning of the stem (see e.g. Şehitoğlu and Bozşahin (1999)).

2. It is not clearly identifiable whether a suffix is a derivational or an inflectional one. This may be because there is no stable definition for a derivational affix though there are some assumptions about that. Stump (1998) states that there are certain criteria to discriminate inflection and derivation and he gives seven of them but he again gives situations and examples to prove that these are not enough to clarify the distinction. These criteria even have conflicts between themselves. Some of these criteria are :

   (a) Part of speech (pos) and semantic change in derivations

   (b) Syntactic behavior of inflections

   (c) Semantic regularity of inflections

   (d) Inflection closes words to further derivation

But these criteria do not constitute a perfect way to discriminate derivation and inflection. There may be cases where an affix satisfies the criteria for both a

derivation and an inflection. For example in Turkish a noun may take inflectional suffixes before '-ki relativization' after which it can be considered as an adjective. This is a derivation after inflection if *-ki* is a derivation (by criteria 1). Secondly there are some really compositional derivations in Turkish. Take *-lAş* suffix in Turkish which makes an intransitive verb out of an adjective. It means to become something, and almost all adjectives may take this suffix and become a verb. *-ArAk* and *-mAdAn* are two other such suffixes which make adverbs out of verbs. They are also very productive and compositional. However this compositionality and predictability is not valid for all suffixes which can be considered as derivational suffixes according to the same classification. This means that there should be levels of derivationality if one insists on building a definition based on these criteria.

(15) a. yabancılaşmak
*'to estrange'*

b. taş-laş-mak
*'to become (like) a stone'*

c. yeşil-leş-mek
*'to become green'*

d. düşün-erek
*'while thinking'*

e. bak-madan
*'without looking'*

Bearing all these in mind, a lexicon structure containing fully inflected words and also some derivations, which are compositional, may be at odds with the computational concerns. According to Hankamer (1996a), without recursion, the verbal morphotactics accepts 1,830,248 forms from one verb root and this number goes up to 9,192,472 for a noun with nominal morphotactics, again without recursion. This means billions of word forms has to be listed in the

lexicon in order to have a comprehensive system. For example the nominal relativization suffix *-ki* in Turkish can create an infinite number of inflected forms from a single noun. Listing these inflected forms for all nouns is not possible. A morphological analyser would reduce the amount of work and the need for storage by formulating the inflected forms instead of listing them.

### 2.1.2 Syntax

Turkish has a certain amount of scrambling among its arguments. A case sytem is crucial for deriving the Predicate Argument Structure (PAS) in such languages. In Turkish the secondary arguments are case marked while the primary argument is in nominative case which is grammatically unmarked. Objects can take either accusative or dative case markers (16a,b).

(16) a. Ahmet kitab-ı oku-du.
Ahmet book-ACC read-PAST
*'Ahmet read the book.'*

b. Ahmet kitab-a bakt-ı.
Ahmet book-DAT look-PAST
*'Ahmet looked at the book.'*

c. Ahmet kitab-ı çocuğ-a ver-di.
Ahmet book-ACC child-DAT give-PAST
*'Ahmet gave the book to the child.'*

Primacy in predicate argument structure is determined by being grammatically less marked (Bozşahin, 1998). Dative marked objects are less primary than accusative marked objects (if exist) in Turkish and nominative (unmarked) is used for subjects in a sentence (16c).

With these properties in mind, Bozşahin (1998) gives case markers the grammatical role of deriving the predicate argument structure. Case marker type raise the nouns to establish the correspondence between the surface constituents and the PAS.

Interestingly, subordinated clauses may also scramble, and the case system, eventually, applies to subordinated clauses (6b). This is morphologically possible only by nominalization of the subordinated verb. *'me(-AGR)'* and *'dHk(-AGR)'* suffixes act as nominalizers on verbs to give way to agreement and case suffixes.

### 2.1.3 Word Order

The canonical word order of Turkish is SOV where V is any kind of predicate and O is object or an adjunct. Turkish is usually considered as a free word order language according to Steele's (1978) classification of languages on word order. Hoffman (1995) claims that SOV is the most common word order , though all six permutations are grammatical, since case-marking rather than word order serves as a guide to construct predicate argument structure in Turkish. She also shows that word order is primarily used to construct the information structure by speakers of Turkish. This is done by placing the previous context at the start of the sentence and focus in immediately preverbal position. She also claims that Turkish cannot be considered as a strictly verb-final language like Japanese.

Underhill (1986, p.16) states that unmarked word order in Turkish is SOV, "however word order is in fact pragmatically controlled and thus appears highly fluid to English speakers." According to Underhill there are two basic principles which restrict the word order:

1. Position immediately to the left of the verb is the focus position.

2. Constituents may move to the right of the verb (thus violating verb final order). Post-verbal position is used for backgrounded, presupposed or afterthought information.

This observation is valuable but it only considers the relation between the information structure and word order restrictions. However tune is also used to mark the focus and background information in Turkish as well as word order. And it is sometimes dominant on word order so these observations are not complete without considering tune as sketched in (17).

(17) a. Ben DÜN kitabı getirdim sen bugun istiyorsun.
*'I brought the book YESTERDAY (but) you want it today.'*

b. Sen bizle GELmiyor musun?
*'Don't you COME with us?*

Another idea is by Kornfilt (1997, p.9). She claims that although Turkish is an SOV language, the basic word order is usually overridden by various other factors.

Kruiff (2001) gives an extensive analysis of word order variations in natural languages. According to his OV-1 hypothesis Turkish is a free word order language since it satisfies the conditions below.

1. It has agreement between the subject and the verb.
2. The strategy to indicate the subject object distinction is case marking and the verb can be inflected with all of tense, aspect, and voice.

On the other hand Bozşahin (2002b) claims that Turkish is a verb-final language, not just SOV. He agrees that Turkish has a flexible word order but he claims that the rightward flexibility is caused by contraposition, not scrambling, which he shows is consistent with post-verbal scrambling and forward gapping.

## 2.2 PC-KIMMO and Two-level Morphology

The Morphological component implemented in this study is based on Koskenniemi's two-level paradigm of morphology (Koskenniemi, 1983), and is developed by the use of the system based on the two-level paradigm, namely PC-KIMMO2 (Antworth, 1995). We first review the essentials of PC-KIMMO and two-level morphology, then we will discuss the computational properties of PC-KIMMO.

Two-level morphology is a general purpose paradigm for morphological description of word structures (Antworth, 1990). Instead of intermediate levels of

representation proposed by Chomsky and Halle (1968), only two levels of representation are proposed (lexical and surface). Rules can be thought of as statements that directly constrain the surface realizations of lexical strings. Each rule would constrain a certain lexical/surface correspondence and the environment in which the correspondence was allowed, required or prohibited.

Two-level morphology is based on three ideas (Kartunnen and Beesley, 2001):

1. Rules are symbol-to-symbol costraints that are applied in parallel, not sequentially like rewrite rules of e.g. Chomsky and Halle (1968)

2. The constraints can refer to the lexical context, to the surface context or to both contexts at the same time.

3. Lexical look-up and morphological analysis are performed concurrently.

PC-KIMMO2 uses this two-level morphology paradigm together with a lexicon and word grammar in order to be able to recognize and generate word structures and provide other linguistic information that a morphological analyser has to provide.

In the rest of this section we are going to elucidate some facilities that PC-KIMMO provides.

### 2.2.1 The Lexicon

The lexicon component of PC-KIMMO is able to provide all the necessary information to the word grammar or to an external system. This is mostly done with features that can be assigned to the lexical entries. Besides this property, PC-KIMMO lexicon also provides information to an FSA that is going to deal with morphotactics, with a property called alternation classes.

A PC-KIMMO lexical entry consists of a phonological form which is the phonological symbol for the mopheme; an alternation class name to be used for tactical constraints, the name of the sublexicon the morpheme belongs to, a features field, values of which will be used either as a guide in morphotactical component or as a provider of some useful information about the word or

1. a:b $\Rightarrow$ L_ R 'Only but not always'
2. a:b $\Leftarrow$ L_ R 'Always but not only'
3. a:b $\Leftrightarrow$ L_ R 'Always and only'
4. a:b $\backslash\Leftarrow$ L_ R 'Never'

Figure 2.3: Rule types for two-level morphology

morpheme which may be used externally, like voice, polarity, gender, syntactic type. For example:

(18)
\lf oku
\lx VERBS
\alt VInfl
\fea tvacc
\gl V

In our system the mechanism for the lexical entries to carry morphosyntactic and semantic information is the feature mechanism. A feature structure is constructed as an output from the values in *\fea* field. These values are the templates for the feature structures defined in the grammar rules file.

### 2.2.2 The Phonological Component

The phonological component of two-level morphology matches the functional representations of morphemes (lexical form) to the orthographic realizations of them. This is done by finite state transducers (FST). Each rule has one of the four forms described in Figure 2.3. PC-KIMMO translates a rule into a FST. The rules indicate the context in which a correspondence is licenced, disallowed or obligatory in the environment specified by the left hand side of the rule. In this terminology $L$ and $R$ are the left and right contexts respectively, and $a : b$ pair is the correspondence of the lexical $a$ to surface $b$.

The FSTs run in parallel to provide a final surface form deducted from the original lexical form. For a detailed analysis of two level rules and phonological component the reader is referred to Antworth (1990).

### 2.2.3 Word Grammar

Word grammar is a context free grammar formalism adapted from PATR (Summer Institute of Linguistics) system to be applied to word structures in order to parse word structure. Word grammar component is added to the second version of PC-KIMMO, namely PC-KIMMO2. Without this word grammar component, it was unable:

1. to provide structural information to the second parties, like parsers. The only output was a sequence of morphemes and a gloss which was quite unable to meet the needs of a representation for the information. This information constitutes : the current part of speech after all the derivations; the features that are owned after going through inflections like: voice, polarity or case; or a feature gained by composition of two morphemes.

2. to get rid of the main cause of inefficiency and redundancy, which is the existence and use of null entities. These are unavoidable when there is a morpheme which is optional, and this usually happens in languages. For example, the plural morpheme is optional before possessive morpheme, and the possessive morpheme is optional before case morpheme in Turkish. For each alternation class that the morpheme is followed by, a null entry is needed and this should be done for all the lexical entries that belong to that sublexicon.

3. to get rid of the duplicate entries that is caused by the discontinuities, that is, a case in which a morpheme requires the existence of another which is not consecutive.

4. to eliminate the bad parses and overgeneration, which is very difficult without a word grammar.

Besides acting as a recognizer, a word grammar also helps to build information structures for the morphological components. These structures may contain

various information with the help of user-defined features. The number and the value of these features are not limited.

To sum up, the idea of word grammar is preferred over the traditional alternation set system because it is more convenient, more efficient, easier to use and more suitable for the computational properties of the morphological system of language processing.

### 2.2.4 Computational Properties

Morphological structures are known to be parsable with finite state methods. Turkish is considered to be a language with concatenative morphology and that kind of morphology can be parsed with simple finite-state grammars (Sproat, 1992, p.44).This means that Turkish morphotactics can be represented as a regular grammar and recognized in linear time; there is no need for a more powerful machine for that.

PC-KIMMO, when it was first implemented aimed to achieve these computational properties. However, Barton (1986) proves that finite-state machinery does not guarantee efficient parsing by reducing the generation problem of PCKIMMO to SAT to show that it is in fact NP-hard when deletions (null symbols) are unrestricted. Koskenniemi and Church's (1988) reply to this critique was that there are no cases in natural languages to satisfy the condition in Barton's reduction, which is having three or more harmony processes. So the exponentiality is never likely to be realized in any practical application. KIMMO system is shown to be efficient with empirical data from Finnish to show linearity between the word length and the number of analysis steps.

## 2.3 Combinatory Categorial Grammars

Combinatory Categorial Grammar (Ades and Steedman, 1982; Steedman, 1985; Steedman, 1987; Steedman, 2000) is an extension to the classical categorial grammar proposed by Ajdukiewicz (1935), and Bar-Hillel (1953) who added directionality. Categorial Grammar (CG) and extensions to it are lexicalist ap-

proaches which deny the existence of a need for movement in syntax or deletion rules. Transparent composition of syntactic structures and semantic interpretations, and flexible constituency property make them desirable and predictive for handling long distance dependencies and non-constituent coordination in many languages (e.g. English, Turkish, Japanese, Irish, Dutch).

There are various extensions to categorial grammars. Steedman (1985) generalizes the function composition rules of Lambek (1958) to capture Dutch long distance dependencies. Composition rules are described in this section. Moortgart (1988), (1997) extends Lambek Calculus with modalities to account for island constraints.

### 2.3.1 CG Formalism

The categories in categorial grammars can be atomic categories or functions which specify the directionality of their arguments. Bozşahin (2002a) represents a lexical item in a CG as the triplet: $\pi - \sigma : \mu$ where $\pi$ is the prosodic element, $\sigma$ is its syntactic type, and $\mu$ its semantic type. Some examples are:

(19) a. *book* $- N$: book

b. *oku* $- (S\backslash NP)/NP$: $\lambda x.\lambda y.\mathsf{read}\, xy$

**Definition 2.1** *Syntactic Types Bozşahin (2002a)*

- *The set of basic syntactic categories:* $\mathcal{A}_s = \{ N, NP, S, S_{-\mathrm{t}}, S_{+\mathrm{t}} \}$
- *The set of complex syntactic categories:* $\mathcal{B}_s$
  - $\mathcal{A}_s \subseteq \mathcal{B}_s$
  - *If* $X \in \mathcal{B}_s$ *and* $Y \in \mathcal{B}_s$, *then* $X \setminus Y$ *and* $X/Y \in \mathcal{B}_s$

In classical CG, there are two kinds of application rules presented below:

(20) Forward Application (>): $X/Y : f \quad Y : a \quad \Rightarrow \quad X : fa$

Backward Application (<): $Y : a \quad X\backslash Y : f \quad \Rightarrow \quad X : fa$

### 2.3.2 Combinatory Categorial Grammars (CCG)

CCG, in addition to functional application rules, has combinatory operators for composition (B), type raising (T), and substitution (S) , which are adopted from Curry's combinatory logic (Curry and Feys, 1958). These operators increase the expressiveness while preserving the transparency of syntax and semantics during derivations. Classical Ajdukiewicz-Bar-Hillel (AB) Grammar is context-free (Bar-Hillel, Gaifman, and Shamir, 1964). After the non-context freeness proof of natural languages (Shieber, 1985), a new formalism and the ways to extend the classical formalism to make it computationally more powerful have been sought. The combinatory rules increase the expressional power of CG to mildly context sensitive which is consistent with the evidence from real linguistic data.[1]

(21) Forward Composition (>B): $X/Y{:}\,f \quad Y/Z{:}\,g \;\Rightarrow\; X/Z{:}\,\lambda x.f(gx)$

Backward Composition (<B): $Y\backslash Z{:}\,g \quad X\backslash Y{:}\,f \;\Rightarrow\; X\backslash Z{:}\,\lambda x.f(gx)$

(22) Forward Type Raising (>T): $X{:}\,a \;\Rightarrow\; T/(T\backslash X){:}\,\lambda f.f[a]$

Backward Type Raising (<T): $X{:}\,a \;\Rightarrow\; T\backslash(T/X){:}\,\lambda f.f[a]$

A generalized form of composition exists. In this form composition operator may be applied several times (n). However this n should be bounded somehow to avoid extra computational power. n is bounded by the valency in the lexicon to retain the Mildly context sensitive power of CCG.

(23) Generalized Forward Composition (> $B^n$):

$X/Y{:}\,f \quad (Y/Z)/\${:}\,...\lambda z.gz... \;\Rightarrow\; X/Z/\${:}\,...\lambda z.f(gz...)$

The convention is called the *$ convention* by Steedman (2000) and is defined recursively as:

[1] Substitution and others will not be mentioned here. Interested reader should refer to Steedman (2000).

(24) *The $ convention*

For a category $\alpha, \{\alpha\$\}$, (respectively,$\{\alpha/\$\}, \{\alpha, \backslash\$\}$) denotes the set containing $\alpha$ and all functions (respectively, leftward functions, rightward functions) into a category in $\{\alpha\$\}$ (respectively, $\{\alpha/\$\}, \{\alpha\backslash\$\}$

Composition and type raising are used to handle syntactic coordination and extraction in languages by providing a means to construct constituents that are not accepted as constituents in other theories. Hoffman (1995) claims that flexible constituency is crucial for handling the scrambling of arguments in languages like Turkish. As Steedman (1991) showed, the spurious ambiguity caused by this flexibility is needed to account for different prosodic bracketings.

### 2.3.3 Computational Power of CCGs

Shieber (1985) proved that natural languages falls into a category more powerful than context-free grammars (CFG) in Chomsky hierarchy using evidence on Swiss-German cross-serial dependencies. This result motivated the search for grammar formalisms that are more powerful than CFGs. Earlier, Bar-Hillel *et. al* (1964) had proved that AB grammars are context-free. Hence they are not expressive enough for natural grammars.

Combinatory rules of CCG extend the capabilities of categorial grammars, and make them mildly context-sensitive. CCG is proved to be weakly-equivalent to other mildly-context sensitive grammars, namely Tree-Adjoining Grammars, Head Grammars and Linear Indexed Grammars (Vijay-Shanker and Weir, 1994).

Another extension of categorial grammars, namely, Categorial Type Logics (CTL) have more power than context-sensitive if there is no restriction on modalities (Carpenter, 1999), but a subset of CTLs were proved to be equivalent to CCGs (Kruijff and Baldridge, 2000).

There has been several variations in CCG mainly to handle word order variation and scrambling efficiently. Multiset-CCG proposed by Hoffman (1995) have more computational power than mild context-sensitivity. The significance of Multiset-CCG is that the set of arguments can be defined rather than one-at-

a-time argument selection. This is shown to work well to handle free word order particularly in Turkish (Hoffman, 1995). Set-CCG is proposed by Baldridge (2000) which has the advantage of handling scrambling in languages more effectively and also being mildly context-sensitive.[2] Underspecification of directionality in Multiset-CCG is not available in Set-CCG. The rationale is that languages are head-consistent in word order, but argument order may vary within a language. For example, the multi-set category $S|\{NP_{nom}, NP_{acc}\}$ for a verb captures all six variations of subject-object-verb (SOV), lexically, hence such a language can not be regarded as verb-final without stipulation. In contrast, the set-CCG category $S\backslash\{NP_{nom}, NP_{acc}\}$ states that all arguments must be to the left of the verb, hence only SOV and OSV are defined lexically and other orders are derived by processes like detopicalization. Bozşahin (2002b) argues that, as far as word order is concerned, being lexical amounts to being basic in linguistic theorization.

[2] Set-CCG is proven to be strongly equivalent to CCG in Baldridge (2000).

# CHAPTER 3

# ARCHITECTURE OF THE MORPHEMIC LEXICON

The architecture of the system is shown in Figure 3.1. The mophological parsing component interacts with the lexicon and analyzes word structures. Morphological component in this study may be thought of as a morphological tagger since the main role of it is to decompose the word into its morphemes and send the parser a list of morpheme labels corresponding to surface morphs (e.g., *-PLU* for *-ler* and *-lar*), together with their morphosyntactic and semantic information, which comes from the lexicon. This information is recognized by the morphemic CCG parser (Bozşahin, 2002a) to make a concurrent construction of syntactic derivation and semantic interpretation.

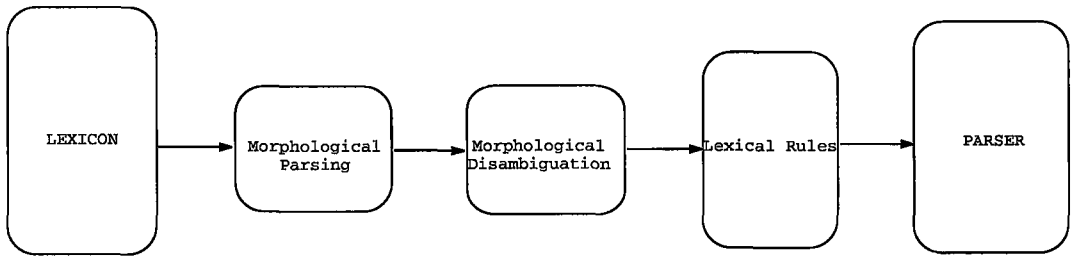


Figure 3.1: Architecture

The system is different from what is proposed in Bozşahin (2002a) (Fig-

ure 1.1(b)) which is taken as a model in this study. The main distinction, the matcher module, is omitted since the morphemes are sent along with their categories and there is no need for a matching process. The lexicon is not split as root and affix lexicons anymore and all the morphemes are treated in the same way by the morphological analyser.

The system has provisions to potentially include a morphological disambiguation module similar to the one in Oflazer and Tür (1996). This disambiguation module would be responsible for finding the most probable morphological analyses for the words depending on the context in which they appear. For example between the two analyses of the word *kitabın* in (25), the disambiguator would select (25a) if the preceding context includes a genitive marker or (25b) if the following context contains a possessively marked object. The system in Oflazer and Tür (1996) is designed to rely on statistical information which is also guided by user defined rules representing the correlations. This results in a considerable improvement in the efficiency since the number of ambiguous parses generally has an exponential impact on the parser. This disambiguator decreases the number of morphological parses per word to 1.10-1.15 on the average from 1.80 average number of parses per word.

(25) a. kitap-POSS2SG *'(your) book'*

b. kitap-GEN *'of the book'*

However, as Oflazer noted, there may be problems in cases like the copula. Example (26) shows the conflict between the determiner reading of *bir*, which is the actual reading of the word in the sentence '*Bu bir masadır*', while the disambiguator sends the adverbial reading which is more likely in this case since the disambiguator sees that the top level pos of the word in the right context is a verb. This is a kind of a mismatch as we mentioned in the introduction. Because *-dir* has to have phrasal scope here but this is prevented by deleting the determiner reading of *bir*. This problem is solved by providing redundant entries in a count table, and having duplicates of the rules for these kinds of

special cases. This solution is both inefficient and requires a tedious work to encounter all possibilities of this kind.

(26) a. bir masadır

b. [[bir masa] -dır]
one/a table -COPULA
*'is a table'*

c. *[bir [masa -dır]]
only/merely(ADV) table -COPULA

The main duty of the lexical rules component in Bozşahin's (2002a) system is to generate the entries of verbs for different lexicalizations of word order. This is needed to handle scrambling. Lexical rules are also needed to handle the changes in the argument structure in processes like causative and passive constructions. For example, causative construction changes the argument structure of the verb as well as the case marking of arguments. The lexical rules component would create the lexical entries for these kinds of iterative processes. The lexical rules for changing the argument structure is not implemented in the current study since the argument structure of verbs is not our main concern.

Bozşahin's (2002a) morphemic CCG parser makes certain simplifying assumptions about the input and some computational processes. These are:

1. Phonological system is not easy to extend since the parser lacks a sound phonology-morphology interface.

2. The input to the parser is assumed to be separated into morphs. This is for the convenience of the parser, which works with morphs rather than words. But this is inconvenient for the user, especially for those who do not have an idea of what the exact morphemes are.

3. Some inherent ambiguity in the input—which has to be preserved— such as in *kitapları*, is lost because of either *kitap-lar-ı* or *kitap-ları* segmentation

before morphemic parsing. Ideally, one would expect two parses for this input, and this can be achieved only if morphological ambiguity is passed on to the parser.

4. The lexical items are searched in the whole lexicon each time a lexical item has to be accessed. Since the current lexicon of the system is a small sample of some basic classes of words, the effect of this matter is invisible. But for a more comprehensive lexicon of, e.g.,thousands of words, the parsing efficiency in terms of speed is expected to be affected negatively. This can be prevented by reducing the number of the lexical entries to be searched, by first sending the words to a morphological analyzer and constructing a sublexicon consisting only of the possible morphemes of the sentence and their categories. The parsing efficiency will no longer be dependent on the size of the lexicon but only on the length of the sentence to be analyzed. This makes a difference especially for languages like Turkish, which has a complex inflectional system.[1]

5. The lexicon has an allomorphic representation; all the morphs of a morpheme are listed in the lexicon as part of a single lexical entry. Lexical access is done by matching the surface form of a particular morpheme. In order to eliminate the list search for this process, the actual lexicon that the parser uses is compiled out of the allomorphic lexicon. This causes a lot of redundancy because apart from the surface structures of these allomorphs there is no difference between them. Assuming each morpheme has at least 2 allomorphs (can go up to 8 for past tense), if a sentence has 3 inflected words and 3 inflections per word, this means that at least (there may also be uninflected words) 21 lexical entries will have to be fetched to parse this sentence. But if functional elements represent the morphemes instead of all allomorphs, this will both help shrink the size of the lexicon and reduce the response time of the parser, which will now

[1] Turkish has 59 inflectional suffixes and a total of 166 bound morphemes according to one estimate (Oflazer, Göçmen, and Bozşahin, 1994).

work on a much smaller sublexicon by which the input is filtered through some morphological processing and morph tagging.

6. Bozşahin (2002a) notes that morphological parsing before lexical access may in fact be more preferable for inflecting languages like Russian and Latin, for which it is difficult to isolate the morphs in the surface form. The affixes are fused to other morphemes in such languages as in Lithuanian (27). [2]

   (27) a. draug-as *'friend (nom.sg)'*

   b. drarug-o *'friend (gen.sg)'*

   c. draug-aĩ *'friend (nom.pl)'*

   d. draug-ũ *'friend (gen.pl)'*

   Instead of having as a separate lexical entres for all these words, we can have it parsed morphologically to obtain the underlying separate morphemes and allow the lexical access from the unique entry for *draug*. The former is required in allomorphic lexical representation, but the latter is feasible in the morphemic lexicon with a morphological parsing component.

## 3.1 Morphological Parser

As Oflazer (1994) noted, two-level systems are used to analyze several languages until now (Alam, 1983; Kartunnen and Wittenburg, 1983; Khan, 1983; Koskenniemi, 1985; Lun, 1983). There are numerous morphological systems implemented for Turkish, too. We will mention only two of them since they are the closest ones to the scope of the current project.

The first one of these is Oflazer (1994). This system has the largest lexicon in terms of both the number of open and closed class words and also the number and variety of the affixes. However this system lacks some basic properties needed to be encountered as a front end to the morphosyntactic parser. The system is

[2] This example is from Bussmann (1996).

designed using the first version of PC-KIMMO environment (Antworth, 1990). It has defects that is mentioned in Section 2.2.3: too many redundant and null entries, and also inadequacy in passing structured information of which a parser may make use, such as feature structures.

The second one is by Pembeci (1998). This system is implemented in PC-KIMMO2 environment and it has a word grammar component. It does not have the inadequacies of information propagation of the earlier systems. However this system has limited capabilities in terms of morphological parsing. The lexicon has 640 open class words (consisting of only verbs and nouns) and 80 inflectional affixes and it is not large enough to be a part of a general purpose NLP system. We will fill the gap in this thesis by extending the noun lexicon considerably, to around 15,000 nouns. The extension of the verbal component requires a labor-intensive work on argument structure of verbs in Turkish, something that is computational equivalent of Levin's (1993) monumental work for English. Although the verbs are parsed correctly on morphological analyser, we leave the issue of creating morphosyntactic categories aside, except to provide a small representative sublexicon of verbs exhibiting various kinds of argument structure (valence, raising and control structures).

It seems that there is a need for a morphological analyzer. The word lexicon of the first system mentioned above is enough for current research purposes and it is also convenient for a more robust system. This lexicon consists of 24,000 morphemes in total. A detailed description of the lexicon is given in Section 3.3.

## 3.2 The Phonological Rules

The phonological rules is one part of the morphological analyser component in the architecture. A comprehensive analysis of morphophonemic processes can be found in Hankamer (1986), Kornfilt (1997), Oflazer (1994), and Öztaner (1996). The phonological rules of the current system is a two-level rules component adapted from Oflazer's (1994) rules. 22 two-level rules cover the most of the morphophonemic processes in Turkish. A detailed desciption of these rules can

be found in Oflazer (1994).

The convention used for Turkish characters is the same as Oflazer's. Turkish diacritics are represented by the capitals. For example *çocuk* will be represented as *Cocuk* and *kapının* will be represented as *kapInIn*. Whenever confusion arises with meta-phonemes, such as *-lAr* to represent underspecification of *a,e* as *A*, we will make it explicit as to whether we are dealing with a diacritic or a meta-phoneme.

## 3.3 The Lexicon

The lexicon of the morphological analysis component from which our structured lexicon is derived was a PC-KIMMO version 1 lexicon originally. It consisted of minimal syntactic information and no semantics, only the lexical form, an alternation class and a gloss to be given as output. This lexicon is converted to PC-KIMMO2 format by using a tool called CONVLEX by Summer Institute of Linguistics (Antworth, 1995). This tool takes a lexical entry of the form:

(28) -nDA CASE-2 '(*CASE* LOC)

which means the lexical form of the morpheme is *-nDA*, its alternation class (the set of morphemes that can follow it) are called CASE-2 and its gloss is '(*CASE* LOC)'. The name of the sublexicon this morpheme belongs to is placed at the top of the list of entries of this kind, and is not shown here. The output of the convlex program for this specific entry will be as in (29). Note that emph\fea feature is missing in this CONVLEX translation. This feature is the main mechanism we added by which we pass syntactic and semantic information.

(29)
```
\lf   -nDA
\lx   RELATIVE
\alt  CASE-2
\fea
\gl   -LOC
```

The only newly introduced word is RELATIVE here which is the sublexicon name which is written as a heading when a new sublexicon starts in PC-KIMMO.

In PC-KIMMO2 this is changed to enable the user to make the grouping based on his purposes, so one does not have to go to that sublexicon to add the new item because the sublexicon field will indicate the sublexicon the entry belongs to.

This lexicon is designed to meet the current needs which are mentioned above. The morphotactics was being handled by a large number of alternation classes. This was not needed anymore because of the word grammar component. Since there is a grammar to parse word structure, the alternation class is only used to indicate which kind of morphemes are *not* allowed after the morpheme at hand. In our system, for example, all lexicons of nominal inflections are in an alternation class called **Infl** which indicates implicitly that a morpheme which is in verbal paradigm cannot follow a morpheme which belongs to a lexicon in this alternation class, but any morpheme in nominal paradigm can follow the current one. This will not cause a problem of overgeneration since word grammar will determine the actual ordering of morphemes and moreover it can prevent some morphemes from combining by putting features disabling the unification of their feature structures. Word grammar will also handle discontinuities very easily which is very difficult and costly in PC-KIMMO version 1.

In current implementation of Turkish morphology that we experimented with, there are 20 lexicon files, grouped according to their part-of-speech, consisting of 24K root word entries (Oflazer, 1994) and suffixes which sums up to about 130 entries.

Some properties of the root lexicon and most of the properties of suffix lexicon are altered. Minor changes have been made to the root lexicon. But the suffix lexicon has been completely redesigned. Alternation classes have been changed. New features are added to help the word grammar and also to form a feature structure. Some of these features contain crucial information for the morphemic parser such as morphosyntactic and semantic categories of morphemes. After alternation classes are merged there was no need for some duplicate entries created as duplicates of the same morpheme since an entry can have only one

alternation class. This can be handled by writing two or more rules for the same suffix in the word grammar.

These features may also be put to use for some phonological processes that cannot be handled in the phonological component. A feature may be put to lexical entries to force the entry to take only one kind of suffix and prevent the others from combining. Take, for example, the aorist allomorphy and causative allomorphy in the verbal paradigm. The unusual characteristic of the aorist morpheme makes it hard to handle with phonological rules because of exceptions—and also exceptions to the exceptions. This process is handled with adding special features to the lexical entries of root words for (30a,b) and to lexical entries of some suffixes for (30c).

(30) Aorist Allomorphy

a. All monosyllabic roots except 13 take *-(A)r*.[3] All compound verbs formed with 'et' verb also take *-(A)r*. The vowel of the aorist suffix raises to a high vowel, namely it becomes *-(H)r*, if :

   - the preceding stem is one of the thirteen exceptional monosyllabic roots.
   - the verb has taken a causative, passive or reflexive suffix (even if the meaning is not compositional anymore)

b. All the other polysyllabic verbs take *-(H)r*.

c. Following the negative morpheme, the aorist suffix becomes *-z* provided that the person suffix following it is not first person (singular or plural). In that case *-z* goes and person suffix *-m* for singular and *-yHz* for plural comes.

[3] This information is from (Hankamer, 1996b).

## 3.4 The Word Grammar

In this section we will concentrate on the two main paradigms of Turkish morphology. Word grammar is the basic module of the morphological analyser component which is responsible for handling these two paradigms and building the feature structure. These paradigms are i) the nominal paradigm and ii) the verbal paradigm. The nominal paradigm applies to all nouns, adjectives, pronouns, and nominalized verbs.

The second paradigm is the verbal paradigm which applies to all verbs. In both paradigms there are some interesting cases and some exceptions to rules that are worth mentioning. For example:

- how to treat compound markers when taking an inflection.
- how adjectives take full nominal inflections.

### 3.4.1 Nominal Paradigm

The nominal morphology (Figure 2.1) is rather regular and more predictable compared to the verbal morphology. Number, possession, case and relativization are the main elements of the nominal morphology. If present, all these suffixes have to be in this relative order. They are optional except when a case suffix must precede a relativization suffix. Also, the possessive marker, the compound marker and the relativizer change the form of the case suffix coming after them. This is interesting because this is not a phonological process; rather it depends on the presence of another morpheme.

(31) a. kapı-DAT → kapıya in absence of a possesive, compound or a relative marker.
*'to the door'*

b. kapı-POSS3SG-DAT → kapısına
*'to his door'*

This leads to two allomorphs of DATIVE suffix *-yA* and *-nA* which are not allomorphs of each other in the traditional sense because they seem to rely on semantic information rather than phonological information while attaching to a word, since possession or relativization is a semantic relation.[4]

There are some suffixes in the verbal paradigm behaving in the same way as the possessive and relativization suffixes in imposing a change in phonology in case morphemes with semantic information, e.g. the negative morpheme. The negative morpheme changes following the aorist morpheme *-Ar* or *-Hr* to *-z* for person suffixes other than first person and it is null in case of the first person.

(32) a. yaz-Ar-lAr
write-AOR-PERS3PL
*'(they) write'*

b. yaz-mA-z-sHn
write-NEG-AOR-PERS2SG
*'(you) don't write'*

c. yaz-mA-0-m
write-NEG-AOR0-PERS1SG
*'(I) don't write'*

It seems that there are two semantic conditions on the aorist morpheme in this case. There are two hypotheses to consider. First, these morphs might not be allomorphs. Then they have to be treated differently. The second one is, there is a problem of modularity here which we describe subsequently.

#### 3.4.1.1 -ki relativization

This is one of the interesting processes in Turkish which makes one think morphological structures can be infinite in principle. One can make indefinitely long words with this process, though they are restricted, pragmatically, to around

[4] Thanks to Cem Bozşahin for helping me discover this.

three levels. Two *-ki* affixes are difficult enough for speakers of Turkish as can be seen in (33).

(33) evdekilerinkiler
house-LOC-REL-PLU-GEN-REL-PLU
*'the ones that belong to the ones who are at home'*

-ki relativizer morpheme also imposes a phonological form on the case morphemes following it. The case morphemes following the relativizer must follow the *(n)A, (n)H* paradigm rather than the *(y)A, (y)H* as in (34).

(34) a. kapı-DA-ki-(n)A
door-LOC-REL-DAT
*'to the one at the door'*

b. kapı-(y)A
door-DAT *'to the door'*

#### 3.4.1.2 Adjectives

The behaviour of adjectives under inflection is interesting. They do not take inflections if they are used as adjectives, but when used as a noun all adjectives can take nominal inflections.[5] This means all adjectives also have the category N in parsing. It is for this reason that the adjectives are mentioned in the nominal paradigm of morphology, not that adjectives can undergo inflections. The morphosyntactic outcome of this property is that adjectives have two categories, which are $N$ and $N/N$.

#### 3.4.1.3 Pronouns

Pronouns can take nominal morphology but there are some minor dissimilarities with the main paradigm, and exceptions which makes it hard to apply all word grammar rules directly to pronouns. For example:

[5] Lewis (1967) collectively calls them '*substantives*' mainly for that reason.

(35) a. bana 'to me'
ben-DAT

b. *benlerde
ben-PLU-LOC

The change in the vowel of the final syllable in (35a) is not usual in nominal paradigm and (35b) is not valid in pronominal morphology though it would be valid with a nominal root. The first example will be validated by an extra entry in the lexicon for *bana* while the second parse will be prevented by putting constraints on the attachment characteristics of pronouns.[6]

#### 3.4.1.4 Compound Nouns

There is a special case for compound nouns in Turkish. Their nominal form is marked with a compound marker and this marker has the same phonological properties as a possessive marker as explained above. The third person possessive form and nominal form of a compound noun are not distinguishable at the level of morphology. Case suffixes follow the rules of nominal paradigm, however, the plural form is different since the plural comes before the compund marker:[7]

(36) a. antepfIstIGI
COMPOUNDN-CM
*'antep pistachio'*

b. antepfIstIklarI
COMPOUNDN-PLU-CM
*'antep pistachios'*

c. *antepfIstIkIlar
COMPOUNDN-CM-PLU

[6] This is done by putting features as contraints that will be used in word grammar.

[7] Göksel (1993) attributes this to the difference in logical (hence semantic) order and morphological order.

All the other inflectional suffixes can combine with compound nouns without a need for modification in the root. But the compound marker is dropped when the compund undergoes a derivation.

(37) antepfIstIkCI *'antep pistachio seller'*

#### 3.4.1.5 Predicative

All categories that can take nominal inflections can be a predicate. The predicative suffixes are NPRES, NCOND, NPAST, and NNARR. After a nominal is made a predicate, it behaves like a verb and it takes person aggreement suffixes. Person aggreement suffixes may attach directly to stems without predicative suffixes. This represents present tense as in (38a).

(38) a. evdeyim
house-LOC-PERS1SG
*'I am home.'*

b. evdedirler
house-LOC-NPRES-PERS3PL
*'They are at home.'*

c. güzelmişsiniz
beautiful-NNARR-PERS2PL
*'You are said to be beautiful.'*

d. yarınkiydi
tomorrow-REL-NPAST
*'It was for tomorrow.'*

e. kitaptakiysen
book-LOC-REL-NCOND-PERS2SG
*'If you are the one in the book.'*

### 3.4.2 The Verbal Paradigm

The verbal paradigm applies to root and derived verbs. The verbal morphology in Turkish is more complex than nominal morphology. The verb can carry tense, person, voice, polarity, aspect, and modality information in several combinations.

(39) oku-mA-mAlH-(y)dH-n
read-NEG-OBLIG-TENSE-PERS2SG
'you shouldn't have read (it)'.

Like the nominal paradigm, the verbal paradigm consists of infinite processes such as causativization. But after 3 levels of recursion the meaning is not interpretable easily as seen in (40c).

(40) a. yap -tır -dı
do -CAUS -PAST
*'she made (someone) do (it).'*

b. yap -tır -t -tı
do -CAUS -CAUS -PAST
*'she made (someone) make (someone) do (it).'*

c. ?yap -tır -t -tır -dı
do -CAUS -CAUS -CAUS -PAST
*'she made (someone) make (someone) make (someone) do (it).'*

## 3.5 The Lexicon-Mophology-Parser Interface

The morphological analyzer discussed in Section 3.1 acts as a front end to a morphemic CCG parser (Bozşahin, 2002a). The flow of information is given in Figure 3.1.

The surface form (input) directly goes to the morphological analyzer where it is decomposed into its morphemes with the help of the morphotactic rules. The resulting structure is a feature structure. Lexical form of the morpheme,

morphosyntactic and semantic information and the functional type of the morpheme are the outputs of the morphological component which are essential for interpretation. Other information internal to the morphotactics component will be filtered. Some typical feature structures are exemplified below.

```
\lf kapI
\lx N
\alt Infl
\fea
\gl N
```

The example above is the lexical entry of a typical noun and examples below are the two inflectional suffixes used in the morphological analysis. The first few features in the *\fea* field are the features used internally by the morphological parser's morphotactics part, while the last one is the feature for the morphosyntactic and semantic category information for a lexical entry of a bound morpheme.

```
\lf -yH
\lx CASE
\alt End
\fea n/n stemsuf acccat
\gl -ACC

\lf -sH
\lx  POSSESSIVE
\alt Infl
\fea n/n 3psg ns/poss poss3cat
\gl -POSS3SG
```

With the lexical entries as above the morphological analyser gives an output as the following for '*kitabI*' which has two parses shown in (25).

```
kitabI
kitab-sH      N-POSS3SG

[....]

N_4:
[ cat:   N
  gloss: N
  head:    [ form:  NounStem
             number:SG
```

```
            pos:   N
            synsem:  [ cat1:    [ res:     [ bcat:    [ name:  n ]
                                                mmod:    [ diac:  b
                                                           type:  lt ] ]
                                  sem:     [ lamterm:kitab ] ] ] ]
  lex:   kitab ]

POSSESSIVE_6:
[ cat:   POSSESSIVE
  gloss: -POSS3SG
  head:    [ from_form:NounStem
             from_pos:N
             synsem:  [ cat1:    [ arg:     [ bcat:    [ gen:   G
                                                         name:  n
                                                         num:   sg
                                                         per:   3 ]
                                                mmod:    [ diac:  n
                                                           type:  lt ] ]
                                  dir:   BACK
                                  res:     [ arg:     [ bcat:    [ name:  n ]
                                                        mmod:    [ diac:  o
                                                                   type:  eq ] ]
                                               dir:   BACK
                                               res:     [ bcat:    [ name:  n ]
                                                          mmod:    [ diac:  o
                                                                     type:  eq ] ] ]
                                  sem:     [ lamterm:F^F ] ]
                      cat2:    [ arg:     [ bcat:    [ name:  n ]
                                                mmod:    [ diac:  b
                                                           type:  eq ] ]
                                  dir:   BACK
                                  res:     [ arg:     [ bcat:    [ def:   minus
                                                                   name:  n ]
                                                        mmod:    [ diac:  m
                                                                   type:  lt ] ]
                                               dir:   BACK
                                               res:     [ bcat:    [ def:   plus
                                                                     name:  n ]
                                                          mmod:    [ diac:  m
                                                                     type:  eq ] ] ]
                                  sem:     [ lamterm:X^Y^comp'X'Y ] ]
                      cat3:    [ arg:     [ bcat:    [ name:  n ]
                                                mmod:    [ diac:  b
                                                           type:  eq ] ]
                                  dir:   BACK
                                  res:     [ arg:     [ bcat:    [ def:   minus
                                                                   name:  n ]
                                                        mmod:    [ diac:  n
                                                                   type:  lt ] ]
                                               dir:   BACK
                                               res:     [ arg:     [ bcat:    [ name:  n ]
                                                                    mmod:    [ diac:  m
                                                                               type:  eq ] ]
                                                            dir:   BACK
                                                            res:     [ bcat:    [ def:   plus
                                                                                  name:  n ]
                                                                       mmod:    [ diac:  m
                                                                                  type:  eq ] ] ] ]
                                  sem:     [ lamterm:X^Y^Z^comp'|comp'X'Y\'Z ] ]
                      cat4:    [ arg:     [ bcat:    [ name:  n ]
                                                mmod:    [ diac:  b
                                                           type:  eq ] ]
                                  dir:   BACK
                                  res:     [ arg:     [ bcat:    [ def:   minus
                                                                   name:  n ]
                                                        mmod:    [ diac:  n
                                                                   type:  lt ] ]
                                               dir:   BACK
```

```
                                        res:     [ arg:     [ bcat:    [ name:  n ]
                                                             mmod:    [ diac:  o
                                                                        type:  eq ] ]
                                                   dir:   BACK
                                                   res:     [ bcat:    [ def:   plus
                                                                         name:  n ]
                                                               mmod:    [ diac:  o
                                                                        type:  eq ] ] ] ]
                                  sem:     [ lamterm:X^Y^|comp'X'Y\ ] ] ]
            to_form:POSSESSIVE
            to_person:P3SG
            to_pos:N ]
  lex:   -sH ]

1 parse found

kitab-yH     N-ACC

[....]

N_4:
[ cat:   N
  gloss: N
  head:    [ form:  NounStem
             number:SG
             pos:   N
             synsem:  [ cat1:    [ res:     [ bcat:    [ name:  n ]
                                                mmod:    [ diac:  b
                                                           type:  lt ] ]
                                  sem:     [ lamterm:kitab ] ] ] ]
  lex:   kitab ]

CASE_5:
[ cat:   CASE
  gloss: -ACC
  head:    [ from_form:NounStem
             from_pos:N
             synsem:  [ cat1:    [ arg:     [ bcat:    [ def:   plus
                                                         name:  n ]
                                               mmod:    [ diac:  o
                                                          type:  lt ] ]
                                  dir:   BACK
                                  res:     [ bcat:    [ case:  acc
                                                         name:  n ]
                                               mmod:    [ diac:  c
                                                          type:  lt ] ]
                                  sem:     [ lamterm:F^F ] ] ]
            to_form:CASE
            to_pos:N ]
  lex:   -yH ]

1 parse found
```

Some important features used in the feature structure are below:

- **gloss** is the functional representation of the morpheme. This is a class name like N(noun) ,PN(pronoun), ADJECTIVE for open class words and like -POSS1SG, -PAST, and -CASE for bound morphemes.

- **head synsem** contains the syntactic and semantic information which will be used by the parser. This part has nothing to do with morphological

```
[ cat:   CASE
  gloss: -ACC
  head:    [ from_form:NounStem
             from_pos:N
             synsem:  [ cat1: [ arg: [ bcat: [ def:   plus
                                               name:  n ]
                                       mmod:    [ diac:  o
                                                  type:  lt ] ]
                                dir: BACK
                                res: [ bcat: [ case:  acc
                                               name:  n ]
                                       mmod:    [ diac:  c
                                                  type:  lt ] ]
                                sem: [ lamterm:F^F ] ] ]
             to_form:CASE
             to_pos:N ]
  lex:   -yH ]
```

Figure 3.2: Feature structure representation for accusative case morpheme

parsing but only passes lexical information to the parser. Only the terminal entries in feature structure tree has this field.

- Each **Cat***i* is a different category for the morpheme and is turned into a separate lexical entry in the sublexicon of the parser. **bcat** and **mmod** contain the agreement and the morphosyntactic modality information of the functor or the argument. For example: $\overset{c}{\triangleleft} N_{acc} \backslash \overset{o}{\triangleleft} N$ is the morphosyntactic category the feature structure in Figure 3.2 represents in Bozşahin's (2002a) notation.[8]

- **lex** is the lexical representation of the morpheme and is turned into a phonological form by the interface and sent to the parser.

This output is turned into a parser recognizable format. This is a sublexicon file that contains the entries of each category (Figure 3.3) ; it is autogenerated by a feature structure parser which is implemented for this purpose.

[8] For a detailed description of the structure of the lexical entries, please refer to Appendix A.

```
sublex(("N":= @s phon "kitab" -- ~n_base$n([_,_,_,_,_,_]):kitab)).

sublex(("-POSS3SG":= @a phon "sH" -- #n_poss$n([_,_,_,_,_,_])\
#n_poss$n([_,_,_,_,_,_])\
~n_num$n([_,_,sg,3,G,_]):F^F)).

sublex(("-POSS3SG":= @a phon "sH" -- #n_comp$n([_,_,_,_,_,plus])\
~n_comp$n([_,_,_,_,_,minus])\
#n_base$n([_,_,_,_,_,_]):X^Y^comp'X'Y)).

sublex(("-POSS3SG":= @a phon "sH" -- #n_comp$n([_,_,_,_,_,plus])\
#n_comp$n([_,_,_,_,_,_])\
~n_num$n([_,_,_,_,_,minus])\
#n_base$n([_,_,_,_,_,_]):X^Y^Z^comp'(comp'X'Y)'Z)).

sublex(("-POSS3SG":= @a phon "sH" -- #n_poss$n([_,_,_,_,_,plus])\
#n_poss$n([_,_,_,_,_,_])\
~n_num$n([_,_,_,_,_,minus])\
#n_base$n([_,_,_,_,_,_]):X^Y^(comp'X'Y))).

sublex(("N":= @s phon "kitab" -- ~n_base$n([_,_,_,_,_,_]):kitab )).

sublex(("-ACC":= @a phon "yH" -- ~n_case$n([_,acc,_,_,_,_])\
~n_poss$n([_,_,_,_,_,plus]):F^F)).
```

Figure 3.3: A sample of autogenerated sublexicon entries

A sentence is decomposed into its words and each word is sent to the morphological parser. The morphological parser outputs a structure similar to the given above for all words and all parses of the words. The sublexicon is in the parser's format and it only contains the possible morphemes in the input and their categories. Only the syntactic processing and interpretation are left to the parser. Since the parser does not deal with the morphophonology of the system this brings a considerable amount of improvement in terms of efficiency.

The parser's output is in the form:

```
>>:
 Morphs  adam kitab -ACC oku -PAST
     PF  adam # kitab - yI # oku - DI
Syn:Sem  s : read'book'man

yes
| ?-
```

which corresponds to a lexical form, a phonological form (PF), category, and a logical form (Syn:Sem).

# CHAPTER 4

# EXPERIMENTAL RESULTS

In this chapter we show that the qualitative improvements of our proposal over Bozşahin's (2002a) model of lexicon do not put too much burden on parsing efficiency.

The experiment apparatus consists of different classes of sentences to test different syntactic structures.We use the same data set as that of Bozşahin (2002a). This data set exemplifies all the morphosyntactic processes that have been shown to pose difficulties in narrow (word) scope of surface attachment versus wide (phrasal) scope of the semantic domain, e.g. word order and case, subordination, relativization, possessives, compounds, varieties of argument structures for verbs. A total of around 200 sentences are included in the data set. The primary goal of our research is not to improve on parsing time results of Bozşahin (2002a) but to provide a general-purpose computational basis for development of combinatory morphemic lexicons for any language, but we expect the qualitative improvements not to affect the parsing performance dramatically. There are theoretical and practical reasons as to why we do not expect to outperform the model. Theoretically, if we commit ourselves to a morphemic grammar, than inflectional morphology is treated just like syntax, and this means context-free processing of inflectional morphology. Although morphology of human languages is claimed to be linear context-free (Creider, Hankamer, and Wood, 1995), the

semantics of inflectional morphology is not necessarily so. Finite-state methods for dealing with the semantics of affixes takes us back to lexemic grammars, which we argued in the introduction to be problematic. The practical reason for slower performance is the size of the lexicon; we have 24000 lexical entries, whereas Bozşahin's (2002a) reported results used around 700.

An example to the qualitative improvements may be better understood by the following example:

Consider the phrase in (41a). The correct interpretation is shown in (41b).

(41) a. çocuğun arkadaşları arabaya vurdu.
*'the child's friends kicked the car.'*

b. child-GEN friend-PLU-POSS3SG car-DAT kick-PAST

c. *child-GEN friend-POSSPL car-DAT kick-PAST

(42) çocuğ un arkadaş ları araba ya vur du.

If the user enters the morphemes in the sequence as in (42), it would lead to an interpretation of morphemes as in (41c). The interpretation is ill-formed since the possessive, which is plural, will not match with the genitive, which is singular. This requires very careful segmentation of morphemes which is hard even for native speakers. However, in the current system the input is given to the system as words, not morphemes, without segmentation which is the job of the morphological analyser.

The problem is more severe in the case of inflecting languages like Russian, Lithuanian and Latin. It is not possible at all to segment the morphemes in this way in inflecting languages since the inflectional morphemes are fused into the stem. The user cannot be expected to provide surface forms of morphemes. Having a separate morphological analyser will solve this problem since the functional representations of the morphemes instead of morphemes themselves will be provided to the parser in such a case.

Table 4.1: The comparison of parsing performances of the current system with Bozşahin's(2002a) Combinatory Morphemic Lexicon (CML). We assume normal form parsing in all cases.

| Sample text type | Number of items in text | | Information about sublexicon | | Avg. CPU time (in msec.) | |
|---|---|---|---|---|---|---|
| | tests | words | numer. per test | lex.ent. per test | CML | current system |
| Word order & case | 44 | 159 | 2 | 13.56 | 392 | 609 |
| Subordination | 14 | 70 | 4 | 28 | 1240 | 3408 |
| Relativization | 23 | 118 | 3.65 | 23.7 | 1347 | 2957 |
| Control verbs | 32 | 147 | 4.13 | 23.3 | 1026 | 3822 |
| Possessives & compounds | 23 | 101 | 3.35 | 17.35 | 684 | 1564 |
| Adjuncts | 13 | 57 | 1.69 | 15 | 633 | 957 |
| *-ki* relatives | 24 | 66 | 1.52 | 17.94 | 273 | 472 |

Table 4.1 shows the parsing times of the tests, the average number of numerations per test and the average size of the sublexicon for a test. All the tests are carried on a Sun Ultra-450 SunOS 5.7 and SICStus 3. The results of Bozşahin's parser with the test sentences are different from what he reported because the current results are observed in a different environment.

The sublexicon of a sentence contains all the categories for a morpheme the sentence contains. Each numeration is a different combination of the different morphological parses of the words in the sentence. The number of numerations is important in the sense that the parsing performance is inversely proportional to the number of numerations since each combination is tried. Since the number of numerations is potentially exponential on the number of morphological parses of the words in the sentence, the number of ambiguous morphological parses of each word affects the total parsing time.

The total number of distinct words that are analysed by the morphological analyser is 137 words. 13 of these do not have a morphological analysis (they are negative examples to test the morphology in Bozşahin's system) and the

parsing of the sentences with these words are prevented in the morphological analysis phase in the current system. The other 124 words have an average morphological parse of 1.41 per word. The average time for the morphological parser to parse a word is 1.14 seconds.

A relatively long sentence consisting of 12 words takes 188 seconds to parse. This sentence has 32 numerations and about 59 sublexicon entries.

Though the current system seems to show less performance in the tests, it should be noted that these results are in a system of around 24000 words. There is a 3-fold increase in the parsing times despite the 32-fold increase in the lexicon size.

The performance in the current system is only proportional to the length of the sentence and is not related with the size of the lexicon of the system after the morphological analysis phase is over. This means the results will not differ much with an even larger lexicon. However, for example in Bozşahin (2002a) the parsing performance is linearly dependent on the size of the lexicon and in a situation that the lexicon contains 15039 nouns, 3282 adjectives and 130 suffixes (and a small sample of verbs) as it does in the current system, its performance is expected to decrease considerably.

# CHAPTER 5

# CONCLUSION

A morphemic lexicon is crucial for natural language parsing because of the conflicts in bracketing of different levels of composition and in cases where the phrasal scope of some morphemes are required by semantics. This is the main reason Bozşahin (2002a) proposes a morphemic lexicon with morphemes containing morphosyntactic information in order to handle phrasal scope of inflections as well as word scope. However, Bozşahin's model has representational problems considering isolating languages in which morphemes are fused into the stem and can not be entered into the lexicon separately, which is a representational requirement for his system. This pre-separation also eliminates the possible ambiguous morphological parses of a word which means an intervention to the system by eliminating the parses which will not be validated until the syntactic and semantic analysis.

We proposed, in this study, a model of interface of syntax, morphology and lexicon to preserve the advantages of a morphemic lexicon while eliminating the deficiencies mentioned above. A morphological analyser is integrated into the system. Handling morphological analysis as a separate module is also more suitable for inflecting languages such as Latin, Russian, Lithuanian. Listing all the inflected forms of the morphemes in such languages is very costly. An approach which is applied to Turkish as in (Bozşahin, 2002a) would be impractical

because morphemes of a word are not easily identifiable looking at the surface form in inflecting languages, unlike agglutinating languages like Turkish. The optimal approach for a morphemic lexicon of an inflecting language is having a morphological analyser handling morphological processes and sending a stream of morphemes to the morphemic parser. The morphological analyser in the system provides a stream of morphemes and their interpretations to the parser. A translator does the work of translation between the morphological analyser's output and the parser's input. The morphophonemic processes are also handled in the morphological analysis phase and this removes the burden of having all allomorphs of a morpheme in the lexicon, which is a considerable improvement in terms of lexicon size for languages like Turkish that have complex phonological processes.

Our experimental results show that a 32-fold increase in the lexicon size and the additional morphological component caused around 3-fold increase in parsing times. This is tolerable if we consider the qualitative improvements gained by having a general-purpose system; the lexicon designer can do her work on lexicon development independent of morphological concerns and preserve the transparency of the syntax-semantics correspondence.

In the current system the different morphological analyses of the words form a numeration for each morphological analysis of each word. This makes the system exponential on the number of parses of the words in the sentence. For a 3-word sentence where each word has 2 different morphological analysis,[1] the numerations to be tried is 8. As part of future work, another mechanism to pass the morphological ambiguity to the parser is crucial for this reason.

The codes for the implementation of the system can be found at ftp://ftp.lcsl.metu.edu.tr/pub/tools/rthesis02.

[1] This is very likely for Turkish.

# APPENDIX A

## How to create lexical entries for the system

Here are some examples on how to create lexical entries for the system.

The lexicon is in PCKIMMO format. The morphosyntactic categories of the lexical entries are created by following the procedure given in this appendix. First of all the category is defined in the word grammar file (grammar.grm) of PCKIMMO-2 as a feature (Figure A.1). The name of this feature is added to the \fea field of the lexical entry after this. Some specifications on how to create the feature structure is the main focus here. All restrictions of PCKIMMO-2 grammar files on the feature names and other things apply to these definitions. Remember to put the name of the feature to the feature list in the main lexicon file.[1]

The value of the lf field of the entry should start with '-' if the entry is a bound morpheme. The features other than plucat in this example are for the use of morphological analyzer.

```
\lf -lAr
\lx PLURAL
\alt Infl
\fea sg/pl n/n ns/ns plucat
\gl -PLU
```

[1] For these kinds of specifications please refer to the PCKIMMO-2 manual.

```
Let plucat be <head synsem> = [cat1: [ arg :[
                                        bcat: [ name: n
                                                num: pl]
                                        mmod: [ diac: b
                                                type: lt]
                                      ]
                                 dir : BACK
                                 res :[
                                        bcat: [ name : n
                                                num: sg ]
                                        mmod: [ diac:n
                                                type: lt]
                                      ]
                                 sem: [lamterm: X^plu'X]
                               ]
                          ]
```

Figure A.1: The word grammar definition of the category for plural morpheme

Figure A.1 is an example feature structure definition for Turkish plural morpheme. The morphemes usually have more than one category. In such cases the category names can go as cat1, cat2 , cat3 etc.[2] The order is not important.

Each category consists of either only a res as in Figure A.3 or res, arg, dir, and sem fields. res is the result category (the functor) arg is the argument and dir is directionality, which can either be BACK or FORW.

Complex categories such as in (43b) are represented as nested feature structures as in Figure A.2. The corresponding entry which the parser recognizes for the complex category in the figure is given in (43a).

(43) a. sublex(("-RELKI":= @a phon "-ki" - #n_relbase$n([_,_,_,_,_,_])
\ (~n_poss$n([_,gen,_,3,_,plus])/ (#n_poss$n([_,gen,N,P,_,_])\
#n_poss$n([5,gen,N,P,_,_]))):X^ (pro^ F)^ and'(poss'pro'X)'F)).

b. -PROki := $\overset{a}{o}$ *ki* – $\overset{l}{\bowtie} N\backslash(\overset{o}{\triangleleft} N/(\overset{o}{\bowtie} N\backslash\overset{o}{\bowtie} N))$: $\lambda x.\lambda f.\mathsf{and}(\mathsf{poss}\,\mathsf{PRO}\,x)(f[\mathsf{PRO}])$

Atomic categories consists of bcat and mmod fields which correspond to the

[2] Please refer to p.43 for an example of multiple categories for a morpheme.

```
Let relkicat be <head synsem> = [
                      cat1 : [ ... ]
                      cat2 :[ ... ]
                      cat3: [ res: [
                                    bcat: [name: n
                                           ]
                                    mmod: [diac: l
                                           type: eq]
                                   ]
                              dir: BACK
                              arg:[ res: [
                                          bcat: [ name: n
                                                  case: gen
                                                  def: plus
                                                  per: 3
                                                 ]
                                          mmod: [diac: o
                                                 type: lt]
                                         ]
                                    dir: FORW
                                    arg:[ res: [
                                                bcat: [ name: n
                                                        case:gen
                                                        num: N
                                                        per: P
                                                       ]
                                                mmod: [diac: o
                                                       type:eq
                                                      ]
                                               ]
                                          dir: BACK
                                          arg: [
                                                bcat: [ name: n
                                                        case:gen
                                                        num: N
                                                        per: P
                                                        obliq: 5
                                                       ]
                                                mmod: [diac: o
                                                       type:eq]
                                               ]
                                         ]

                                   ]
                             sem: [lamterm: X^|pro^F\^and‘|poss‘pro‘X\‘F ]
                            ]
                     ]
```

Figure A.2: An example of a complex morphosyntactic category

```
Let PRONOUNS be
                <head pos>= PRO
                <head number>=!SG
                <head form>= POSSESSIVE
                <head synsem> = [cat1: [ res: [ mmod:[diac: b
                                                      type: lt ]
                                                bcat: [name: n]
                                              ]
                                        ]
                                ]
                <head synsem cat1 sem lamterm> = <lex>
```

Figure A.3: The word grammar definition of the category for pronouns

agreement and morphosyntactic modalities of the category. The following restrictions apply to bcat and mmod.

1. mmod can be a variable or a feature structure consisting of 2 fields: diac (diacritic) and type (modality type). diac can be one of the 18 in Table A.1 for Turkish. type is either lt (less than or equal) which corresponds to $\lhd$ in Bozşahin's(2002a) notation, or eq (equal) which corresponds to $\bowtie$.

2. bcat can be a variable or a feature structure. The features of bcat are the type name (N, S or NP), which is represented by name, and the attributes of this type e.g., num (number), gen (gender), obliq (obliqueness). The values of these attributes are not restricted and are translated into a structure like the following. The unspecified categories are left blank (_)
   #n_poss$n([5,gen,N,P,_,_])
   #n_poss is the parser's representation for $\overset{5}{\bowtie} N$.

The sem field represents the semantics of the category. Each category of each morpheme has to have a sem field. The following restrictions apply for semantics field.

1. It may consist of only a lamterm feature or lamterm and p.

Table A.1: The diacritics for Turkish

| Category N (or NP) | | Category S | |
|---|---|---|---|
| free | f | s-person | s |
| n-case | c | s-modal | d |
| n-comp | m | s-tense | t |
| n-poss | o | s-abil | a |
| n-num | n | s-neg | g |
| n-base | b | s-imp | i |
| n-relbase | l | s-pass | p |
| n-root | r | s-caus | u |
| | | s-reflex | x |
| | | s-recip | r |
| | | s-base | v |

2. p feature is used for automatic generation of the predicate names in cases like the lexical form is the name of the predicate (similar to the use of lex in Figure A.3 but is used in cases like x^y^read'x'y where there are variables in lambda expression.

3. The parentheses in the lambda expressions are '|' for '(', and '\' for ')' since '(' and ')' characters are reserved in PCKIMMO.

# REFERENCES


Ades, A.E. and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Ajdukiewicz, Kazimierz. 1935. Die syntaktische konnexitat. In *Polish Logic, ed. Storrs McCall, Oxford University Press*, pages 207–231.

Alam, Yukiko Sasaki. 1983. A two-level morphological analysis of Japanese. *Texas Linguistic Forum*, 22:229–252.

Antworth, Ewan L. 1990. *PC-KIMMO: A two-level Processor for Morphological Analysis.* Summer Institure of Linguistics, Dallas.

Antworth, Ewan L. 1995. *PC-KIMMO Version 2.* Summer Institute of Linguistics.

Aronoff, M. 1976. *Word Formation in Generative Grammar.* MIT Press.

Baldridge, Jason M. 2000. Strong equivalence of CCG and Set-CCG. ms. University of Edinburgh, 2000.

Bar-Hillel, Yehoshua, C. Gaifman, and E. Shamir. 1964. On categorial and phrase structure grammars. In *Language and Information ed. Bar-Hillel, Addison-Wesley*, pages 99–115.

Bar-Hillel, Yeoshua. 1953. A quasi-arithmetic description for syntactic description. *Language*, 29:47–58.

Barton, G. Edward. 1986. Computational complexity in two-level morphology. In *ACL Proceedings of 24th Annual Meeting.*

Beard, R. 1995. *Lexeme-Morpheme Base Morphology.* State University of New York Press.

Bozşahin, Cem. 1998. Deriving the predicate-argument structure for a free word order language. In *Proceedings of COLING-ACL'98, Montreal*, pages 167–173.

Bozşahin, Cem. 2002a. The combinatory morphemic lexicon. *Computational Linguistics*, 28(2):145–186.

Bozşahin, Cem. 2002b. Lexical origins of word order and word order flexibility. ms. Middle East Technical University, 2002.

Bozşahin, Cem and Elvan Göçmen. 1995. A categorial framework for composition in multiple linguistic domains. In *Proceedings of the Fourth International Conference on Cognitive Science of NLP, Dublin.*

Bussmann, Hadumod. 1996. *Routledge Dictionary of Language and Linguistics.* Routledge, London.

Carpenter, Bob. 1999. The Turing-completeness of multimodal categorial grammars. In *Papers presented to Johan van Benthem in Honor of his 50th Birthday, ESSLLI, Utrecht.*

Chomsky, Noam. 1959. On certain formal properties of grammars. *Information and Control*, 2(2):137–167.

Chomsky, Noam. 1995. *The Minimalist Program.* MIT Press, Cambridge, Mass.

Chomsky, Noam and M. Halle. 1968. *The Sound Pattern of English.* Harper and Row.

Creider, Chet, Jorge Hankamer, and Derick Wood. 1995. Preset two-head automata and natural language morphology. *International Journal of Computer Mathematics*, 58:1–18.

Crystal, David. 1998. *A dictionary of Linguistics and Phonetics.* Blackwell.

Şehitoğlu, Onur and Cem Bozşahin. 1999. Lexical rules and lexical organization: Productivity in the lexicon. In *Breadth and Depth of Semantic Lexicons, ed. Evelyn Viegas.* Kluwer.

Curry, Haskell B. and Robert Feys. 1958. *Combinatory Logic: Vol.1.* North Holland.

Göksel, Aslı. 1993. *Levels of Representation and Argument Structure in Turkish.* Ph.D. thesis, SOAS.

Güngördü, Zelal and Kemal Oflazer. 1995. Parsing Turkish using the Lexical Functional Grammar. *Machine Translation*, 10:293–319.

Hankamer, Jorge. 1986. Finite state morphology and left to right phonology. In *Proceedings of the West Coast Conference on Formal Linguistics.*

Hankamer, Jorge. 1996a. Morphological processing and the lexicon. In *Lexical Representation and Process ed. William Marslen-Wilson*, pages 392–408. MIT Press.

Hankamer, Jorge. 1996b. Phonological rules of Turkish. Talk handout, METU Ankara, 1996.

Hoffman, Beryl. 1995. *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish.* Ph.D. thesis, University of Pennsylvania.

Kartunnen, Lauri and Kenneth R. Beesley. 2001. A short history of two-level morphology. In *ESSLLI 2001 Lecture Notes*, August.

Kartunnen, Lauri and K. Wittenburg. 1983. A two-level morphological analysis of English. *Texas Linguistic Forum*, 22:217–228.

Khan, Robert. 1983. A two-level morphological analysis of Rumanian. *Texas Linguistic Forum*, 22:253–270.

Kornfilt, Jaklin. 1997. *Turkish.* Routledge.

Koskenniemi, Kimmo. 1983. Two-level model for morphological analysis. In *International Joint Conference on Artificial Intelligence,IJCAI-83*, pages 683–685.

Koskenniemi, Kimmo. 1985. An application of the two-level model to Finnish. In *Computational morphosyntax: a report on research 1981-1984, ed. Fred Karlsson.* University of Helsinki Department of General Linguistics.

Koskenniemi, Kimmo and Kenneth W. Church. 1988. Complexity, two-level morphology and finnish. In *COLING-88.*

Kruijff, Geert-Jan. 2001. Competence and performance modelling of free word order. In *ESSLLI 2001 Lecture Notes*, August.

Kruijff, Geert-Jan and Jason M. Baldridge. 2000. Relating categorial type logics and CCG through simulation. ms. University of Edinburgh, 2000.

Lambek, Joachim. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170.

Levin, B. 1993. *English Verb classes and alternations: a preliminary investigation.* University of Chicago Press, Chicago.

Lewis, G.L. 1967. *Turkish Grammar.* Oxford University Press, Oxford.

Lieber, Rochelle. 1980. *On the Organization of the Lexicon.* Ph.D. thesis, MIT.

Lun, S. 1983. A two-level morphological analysis of French. *Texas Linguistic Forum*, 22:271–278.

Matthews, P.H. 1972. *Inflectional Morphology: a theoretical study based on aspects of Latin verb conjugations.* Cambridge University Press.

Matthews, P.H. 1997. *A Concise dictionary of Linguistics.* Oxford.

Moortgat, Michael. 1988. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris.

Moortgat, Michael. 1997. Categorial type logics. In *Handbook of Logic and Langauge, eds.Johan van Benthem and Alice ter Meulen.* MIT Press.

Muller, Stefan. 1999. *Deutsche Syntax deklarativ.Head-Driven Phrase Structure Grammar fur das Deutsche.* Linguistische Arbeiten.

Oflazer, Kemal. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 6(2).

Oflazer, Kemal, Elvan Göçmen, and Cem Bozşahin. 1994. An outline of Turkish morphology. Technical Report TU-LANGUAGE, NATO Science Division SfS III, Brussels.

Oflazer, Kemal and Gökhan Tür. 1996. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In Eric Brill and Kenneth Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Somerset, New Jersey, pages 69–81.

Öztaner, Serdar Murat. 1996. A word grammar of Turkish with morphophonemic rules. Master's thesis, Computer Engineering Department, Middle East Technical University.

Pembeci, Izzet. 1998. A unification-based tool for learning of Turkish morphology. Master's thesis, Computer Engineering Department, Middle East Technical University.

Pesetsky, D. 1979. Russian morphology and lexical theory. Master's thesis, MIT.

Pesetsky, D. 1985. Morphology and logical form. *Linguistic Inquiry*, 16:193–246.

Sadock, J.M. 1991. *Autolexical Syntax: a theory of parallel grammatical representations.* University of Chicago Press, Chicago.

Shieber, Stuart. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

Spencer, Andrew. 1991. *Morphological Theory.* Blackwell.

Sproat, Richard. 1985. *On Deriving the Lexicon.* Ph.D. thesis, MIT.

Sproat, Richard. 1992. *Morphology and Computation.* The MIT Press.

Sproat, Richard. 1998. Morphology as component or module. In *The Handbook of Morphology, eds. Andrew Spencer, Arnold M.Zwicky*, pages 335–348. Blackwell.

Steedman, Mark. 1985. Dependencies and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Steedman, Mark. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Steedman, Mark. 1991. Structure and intonation. *Language*, 67:260–296.

Steedman, Mark. 2000. *The Syntactic Process.* MIT Press.

Steele, Susan. 1978. Word order variation. In *Universals of Language Volume 4 : Syntax, eds Joseph H. Greenberg*, pages 585–624.

Stump, Gregory T. 1991. A paradigm-based theory of morphosemantic mismatches. *Language*, 67:675–725.

Stump, Gregory T. 1998. Inflection. In *The Handbook of Morphology, eds. Andrew Spencer, Arnold M.Zwicky*, pages 13–43. Blackwell.

Underhill, Robert. 1986. Turkish. In *Studies in Turkish Linguistics, eds Slobin, Zimmer*, pages 2–23. John Benjamins Pubishing Company.

Vijay-Shanker, K. and David Weir. 1994. The equivalence of the four extensions of context-free grammar. *Mathematical Systems Theory*, 27:511–546.

Williams, E. 1981. On the notions 'lexically related' and 'head of a word'. *Linguistic Inquiry*, 12:245–274.