DESIGN AND IMPLEMENTATION OF A COMPUTATIONAL LEXICON FOR TURKISH

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND INFORMATION SCIENCE
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Abdullah Kurtuluş Yorulmaz

February, 1997

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

Asst. Prof. Kemal Oflazer(Advisor)

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

Assoc. Prof. Halil Altay Güvenir

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

Asst. Prof. İlyas Çiçekli

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray, Director of Institute of Engineering and Science

ABSTRACT

DESIGN AND IMPLEMENTATION
OF
A COMPUTATIONAL LEXICON
FOR TURKISH

Abdullah Kurtuluş Yorulmaz
M.S. in Computer Engineering and Information Science
Supervisor: Asst. Prof. Kemal Oflazer
February, 1997

All natural language processing systems (such as parsers, generators, taggers) need to have access to a lexicon about the words in the language. This thesis presents a lexicon architecture for natural language processing in Turkish. Given a query form consisting of a surface form and other features acting as restrictions, the lexicon produces feature structures containing morphosyntactic, syntactic, and semantic information for all possible interpretations of the surface form satisfying those restrictions. The lexicon is based on contemporary approaches like feature-based representation, inheritance, and unification. It makes use of two information sources: a morphological processor and a lexical database containing all the open and closed-class words of Turkish. The system has been implemented in *SICStus Prolog* as a standalone module for use in natural language processing applications.

Key words: Natural Language Processing, Lexicon

ÖZET

TÜRKÇE İÇİN
BİR HESAPSAL SÖZLÜĞÜN TASARIMI VE
GERÇEKLEŞTİRİLMESİ

Abdullah Kurtuluş Yorulmaz
Bilgisayar ve Enformatik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. Kemal Oflazer
Şubat, 1997

Bütün doğal dil işleme sistemleri (örneğin çözümleyiciler, üreticiler, metin işaretleyiciler) dildeki kelimeler hakkında, bir sözlüğe erişmeye ihtiyaç duyarlar. Bu tezde, Türkçe'de doğal dil işleme için bir sözlük mimarisi sunulmuştur. Bir kelimenin yüzeysel hali ve kısıtlayıcı diğer özellikler içeren sorguya karşılık, sözlük, verilen kelimenin yüzeysel halinin, bu kısıtlayıcı özellikleri sağlayan her çözümü için biçimbirimsel/sözdizinsel, şekilsel ve anlamsal özellikler içeren bir özellik yapısı üretir. Sözlük, özellik temelli temsil, kalıtım ve birleştirme gibi çağdaş yaklaşımlara dayanır. İki bilgi kaynağı kullanır: bir sözcükyapısal işleyici ve Türkçe'nin bütün açık ve kapalı kelime gruplarını içeren bir kelime veritabanı. Sistem, *SICStus Prolog*'da kendi başına çalışabilecek ve doğal dil işleme uygulamalarında kullanılabilecek şekilde gerçekleştirilmiştir.

Anahtar sözcükler: Doğal Dil İşleme, Sözlük

ACKNOWLEDGMENTS

I am very grateful to my supervisor, Asst. Prof. Kemal Oflazer, for his invaluable guidance, motivation, and patience during the development of this thesis. It was a real pleasure to work with him.

I would like to thank Assoc. Prof. Halil Altay Güvenir and Asst. Prof. İlyas Çiçekli for reading and commenting on the thesis.

I owe special thanks to my colleagues Dilek Z. Hakkani and Gökhan Tür and other friends Yücel Saygın, Uğur Çetintemel, Gamze D. Tunalı and Murat Bayraktar for their endless intellectual and moral support during my graduate study.

Finally, I would like to express my deepest gratitude to my parents for their infinite moral support and patience to me. I dedicate this thesis to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language processing (NLP) is a research area, under which the aim is to design and develop systems to process, understand, and interpret natural language. It employs knowledge from various fields like artificial intelligence (in knowledge representation, reasoning), formal language theory (in language analysis, parsing), and theoretical and computational linguistics (in models of language structure).

There are many applications of NLP such as translation of natural language text from one language to another, interfacing machines with speech or speech-to-speech translation, natural language interfaces to databases, text summarization, text preparation aids such as spelling and grammar checking/correction, etc.

One of the first applications of NLP is machine translation (MT). The research was funded by military and intelligence communities. These systems, what we call *first generation*, translate text almost word by word; the result was a failure. But considering the lack of theories, methods, and resources with semantics and ambiguities in natural language text, the result is not surprising [4]. [1] Today with the advance of theories, resources, etc., MT is not a dream; even there are MT systems available in the market.

Many components of NLP systems, like syntactic analyzers, text generators, taggers, and semantic disambiguators, need knowledge about words in the language. This information is stored

[1] Consider the following well-known utterance:

(1) a. Time flies like an arrow.

b. Fruit flies like a banana.

The ambiguity in the sentences above can be resolved by utilizing the knowledge: *fruit flies* is a meaningful phrase but *time flies* is not. However, even today, most systems cannot access this kind of information.

in the *lexicon*, which is becoming one of the central components of all NLP systems.

In this thesis, we designed and implemented a computational lexicon for Turkish to be employed in an MT project, which aims to develop scientific background and tools to translate computer manuals from Turkish to English and vice versa (see Figure 1.1 for a simplified architecture of this system).

A similar work for this project is the design and implementation of a verb lexicon for Turkish by Yılmaz [16]. This lexicon contains only verb entries to be utilized in syntactic analysis and verb sense disambiguation.

Our work aims to develop a generic lexicon for Turkish, which can provide morphosyntactic, syntactic, and semantic information about words to NLP systems. The lexicon contains entries for all lexical categories of Turkish with the information content also covering the Yılmaz's work. The morphosyntactic information is not directly encoded in the lexicon, rather obtained through a morphological analyzer integrated into the system.

The development of our work is carried out in two steps:

1. determining the lexical specification for each of the lexical categories of Turkish, that is morphosyntactic, syntactic and semantic phenomena to be encoded in the lexicon,

2. developing a standalone system that will provide the encoded information to NLP systems for a given input.

In this thesis, we present design and implementation of such a lexicon.

The outline of the thesis is as follows: In Chapter 2, we introduce the concept of lexicon with examples from related work. In Chapter 3, we present a comprehensive categorization for Turkish lexical types and associated lexical specification. Next chapter gives the operational aspects of our lexicon, that is the interface of the system and algorithms used in producing the result. In Chapter 5, we go through the implementation of the system and give sample runs. Chapter 6 concludes and gives suggestions.

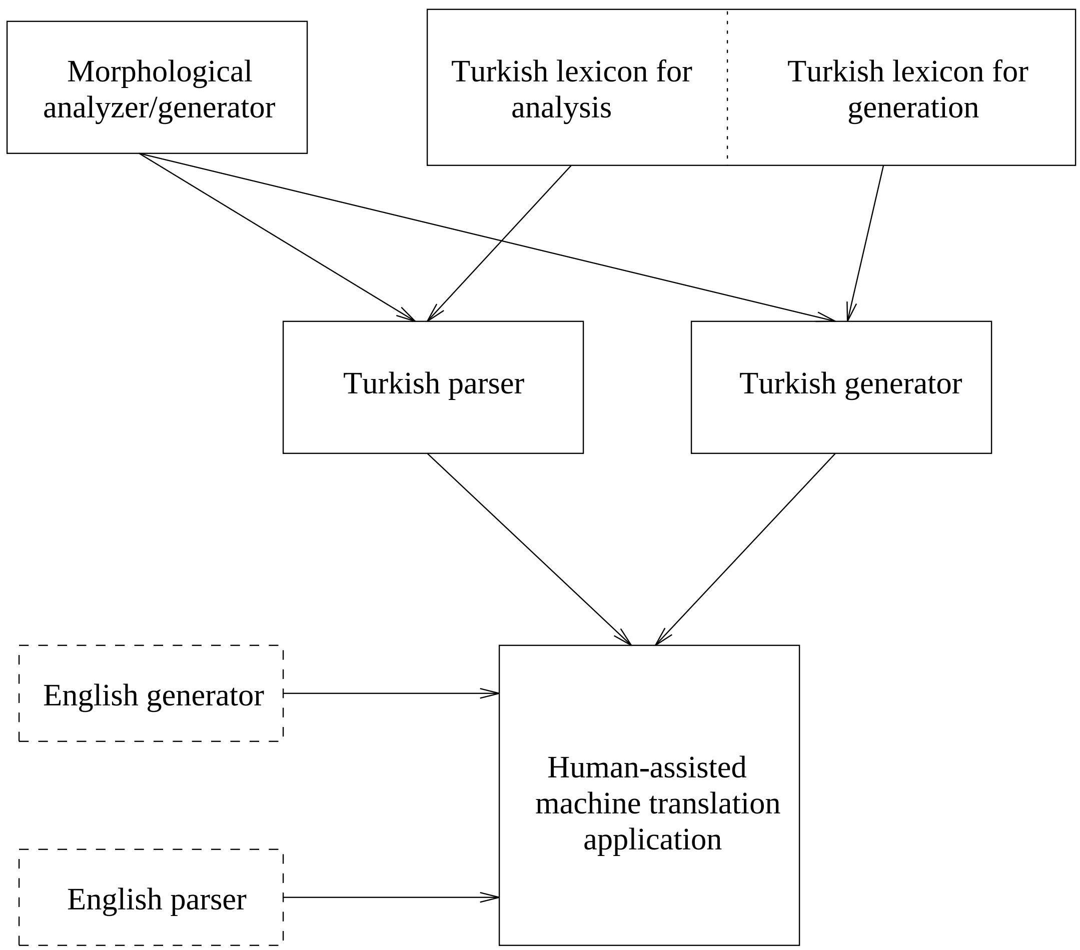

Figure 1.1: Simplified architecture of the MT system that would use our lexicon.

Chapter 2

The Lexicon

Lexicon is the collection of morphological/morphosyntactic, syntactic and semantic information about words in the language. It has been a critical component of all NLP systems as they move from toy system operating in demonstration mode to real world applications requiring wider vocabulary coverage and richer information content.

In this chapter, we will first briefly introduce the concept of lexicon and the need for it. Then, we will give the role of lexicon in NLP with specific examples from syntactic analysis and verb sense disambiguation. Finally, we will present an example work, which is on reaching a common lexical specification in the lexicon among European languages.

2.1 Lexicon

For a long time the lexicon was seen as a collection of idiosyncratic information about words in the language. As the requirements of NLP systems, which perform various tasks ranging from speech recognition to machine translation (MT) in wide subject domains, grow, those systems need larger lexicons. Even simple applications such as spelling checkers may require morphological, orthographic, phonological, syntactic, and semantic information (for disambiguation) with realistic vocabulary coverage [1]. For instance, The Core Language Engine, which is a unification-based parsing and generation system for English, has a lexicon containing 1800 senses of 1200 words and phrases [2]. Thus, the lexicon design and development has become the one of the central issues for all NLP systems.

There are two ways to develop the information content of a lexicon: hand-crafting and use of machine-readable resources. The first is the classical and costly way of developing the content. However, there is a growing trend to use existing machine-readable resources, such as electronic dictionaries and text corpora, to derive useful information. Research in this area has yielded significant results in extracting morphosyntactic and syntactic information, but the results in semantic information side are not yet satisfactory [10].

2.2 The Role of Lexicon in NLP

NLP systems need to access lexical knowledge about words in the language. This information can be morphosyntactic, such as stem, inflectional and derivational suffixes (by means of listing them explicitly or generation), syntactic, such as grammatical category and complement structures, and semantic, such as multiple senses and thematic roles. Depending on the NLP task being performed, other information can be utilized such as mapping between lexical units and ontological concepts for transfer tasks in MT, text planning information for generation, orthographic and phonological information for speech processing applications.

In the following two sections, we will describe the role of lexicon in syntactic analysis and verb sense disambiguation.

2.2.1 The Role of Lexicon in Syntactic Analysis

The following paragraph is taken from Zaenen and Uszkoreit [17], which briefly describes text analysis:

"We understand larger textual units by combining our understanding of smaller ones. The main aim of linguistic theory is to show how these units of meaning arise out of the combination of the smaller ones. This is modeled by means of a grammar. Computational linguistics then tries to implement this process in an efficient way. It is traditional to subdivide the task into syntax and semantics, where syntax describes how the different formal elements of a textual unit, most often the sentence, can be combined and semantics describes how the interpretation is calculated."

The grammar consists of two parts: a set of rules describing how to combine small textual units into larger ones, and a lexicon containing information about those small units. In recent theories of grammar, the first part is reduced to one or two general principles, and the rest of the information is encoded in the lexicon.

Now we will briefly describe the analysis lexicon in KBMT-89 system [5]. KBMT-89 is a knowledge-based machine translation system, in which source language text is analyzed into a language independent representation (namely *interlingua*) and generated in the target language.

There are two other methods used in MT other than interlingua method: *direct* and *transfer* *method*. In the former one, the source text is directly translated to target language, almost word by word with some arrangements, however, in the second one source text is analyzed into an abstract representation, which is then transfered into another abstract representation for the target language, and finally generated as the target language text. Knowledge-based MT requires more syntactic and semantic information, so a larger and richer lexicon, than the other methods, such as language independent knowledge-base for modeling the subworld of translation, etc.

Knowledge acquisition in KBMT-89 is manual, but aided with special tools so that partial automation is achieved. KBMT-89 uses three types of lexicon:

1. *concept lexicon*, which stores semantic information for parsing and generation,
2. *generation lexicon*, which contains information for the open-class words (e.g., nouns, which accept new words in time), in the target language (in that special case, it is Japanese), and
3. *analysis lexicon*, which stores morphological and syntactic information, word-to-concept mapping rules, and information for the mapping case role structures (thematic roles) to subcategorization patterns.

Each entry in the analysis lexicon contains the following information: a word, its syntactic category, inflection, root-word form, syntactic features, and mappings. Syntactic features and mappings can be specified locally or through inheritance by properly setting a pointer to a class in the syntactic feature or structural mapping hierarchy.

Here are two example entries from the English analysis lexicon for the verb and noun interpretations of *note*:

```
(‘‘note’’ (CAT V)
    (CONJ-FORM INFINITIVE)
    (FEATURES
        (CLASS CAUS-INCHO-VERB-FEAT)
        (all-features
            (*OR*
                ((FORM INF) (VALENCY (*OR* INTRANS TRANS)) (COMP-TYPE NO)
                 (ROOT NOTE))
                ((PERSON (*OR* 1 2 3)) (NUMBER PLURAL) (TENSE PRESENT)
                 (FORM FINITE) (VALENCY INTRANS TRANS)
                 (COMP-TYPE NO) (ROOT NOTE))
```

```
                ((PERSON (*OR* 1 2)) (NUMBER SINGULAR) (TENSE PRESENT)
                 (FORM FINITE) (VALENCY INTRANS TRANS))
                 (COMP-TYPE NO) (ROOT NOTE))))
    (MAPPING (local
                 (HEAD (RECORD-INFORMATION)))
             (CLASS AG-TH-VERB-MAP)))
```

In the frame above, first three slots give the headword, its category and word form, that is *note*, *verb* and *infinitive*, respectively. The next slot, `FEATURES`, gives the syntactic features by inheriting the features of the class `CAUS-INCHO-VERB-FEAT`, which are the features of *causative-inchoative* verb class, and adding other features locally, such as valence, root word form, and agreement marker in each of the three cases, as arguments of `*OR*`. The last slot, `MAPPING`, gives word-to-concept mapping, that is the verb *note* is mapped to the ontological concept `RECORD-INFORMATION` in the concept lexicon, and mapping of case role structures to subcategorization patterns by inheriting from `AG-TH-VERB-MAP` class in the structural mapping hierarchy, which is the mapping for *agent-theme* verbs.

```
(``note'' (CAT N)
    (CONJ-FORM SINGULAR)
    (FEATURES
        (CLASS DEFAULT-NOUN-FEAT)
        (all-features
            (PERSON 3) (NUMBER SINGULAR) (COUNT YES) (PROPER NO)
            (MEAS-UNIT NO) (ROOT NOTE)))
    (MAPPING
        (local
            (HEAD (MENTAL-CONTENT)))
        (local
            (HEAD (TEXT-GROUP (CONVEY (COMMUNICATIVE-CONTENT)))))
        (CLASS OBJECT-MAP)))
```

The frame above states that the noun *note* is *singular*, inherits all the syntactic features of the class `DEFAULT-NOUN-FEAT` in addition to its local features; for example its agreement marker is *3sg*, it is countable and not a proper noun. The `MAPPING` slot gives its mapping to the entries in the concept lexicon, that is *note* describes a mental content or a text group conveying a communicative content. It also inherits all the word-to-concept mappings of the class `OBJECT-MAP`.

2.2.2 The Role of Lexicon in Verb Sense Disambiguation

The second specific usage of the lexicon that we will describe is in verb sense disambiguation specifically for Turkish due to the work by Yılmaz [16].

Verb is the most important component in the sentence; it gives the predicate. Thus, resolving lexical ambiguities concerning the verb is very important in syntactic analysis, especially in MT. There are three kinds of lexical ambiguities:

1. *polysemy*, in which case a lexical item has more than one senses close to each other, as in *para ye-* (*cost a lot of money*) and *kafayı ye-* (*get mentally deranged*). For example, *Türk Dil Kurumu Dictionary* gives 40 senses for the verb *çık* and 32 senses for the verb *at*.
2. *homonymy*, in which case the words have more than one interpretation having no obvious relation among them, e.g., *vurul-* has two interpretations: *fall in love with* and *be wounded*.
3. *categorical ambiguity*, in which case the words have interpretations belonging to more than one category, as in *ek* (noun, *appendix/suffix*) and (verb, *sow*).

The claim in Yılmaz's work is that by trying to match the morphological, syntactic, and semantic information in the sentential context of a verb (i.e., the information in its complements) with the corresponding information of the verb entries in the lexicon, the correct interpretation and sense of the verb can be determined. For instance, consider the following example:

(2) a. Memur para yedi.
`official money accept bribe+PAST+3SG`
'The official accepted bribe.'

b. Araba çok para yedi.
`car a lot of money cost+PAST+3SG`
'The car costed a lot.'

In the sentences above, the verb *ye-* is used in two different senses as *accept bribe* and *cost a lot*. The encoding in the lexicon for the first sense states that the head of the direct object's noun phrase is *para* with no possessive or case marking, and the subject is human. For the second sense, the head of the direct object's noun phrase is *para* and the subject is non-human. By applying those constraints, the correct interpretation can be determined. In the application of semantic constraints, however, an ontology (i.e., knowledge-base, which describes the objects, events, etc. in a subject domain) for nouns should be utilized, for example, in testing whether *memur* is human or not.

The lexicon consists of a list of entries for verbs. Each entry is identified with its headword, and contains a list of argument structures, in which there are the labels of the arguments, morphological, syntactic, and semantic constraints, and a list of senses associated with those argument structures. Each sense has another set of constraints specific for that sense and some descriptive information, such as semantic category, mapping of thematic roles to subcategorization patterns, concept name, etc.

Below, we provide the lexicon entry for the verb *ilet-*, which has two argument structures and three senses (i.e., *conduct*, *convey*, and *tell*). In order to save space, we omit the second argument structure and the last sense associated with it. Here is the lexicon entry for *ilet-*:

```
((HEAD . "ilet")
 (ENTRY
     (ARG-ST1
        (ARGS
            (SUBJECT
                (LABEL . S)
                (SEM . T)
                (SYN OCC S OPTIONAL)
                (MORPH . T))
            (DIR-OBJ
                (LABEL . D)
                (SEM . T)
                (SYN OCC D OBLIGATORY)
                (MORPH
                    (OR
                        (1 CASE D NOM)
                        (2 CASE D ACC)))))
        (SENSES
            (SENSE1
                (CONST POWER-ENERGY-PHYSICALOBJECT D)
                (V-CAT PROCESS-ACTION)
                (T-ROLE
                    (1 AGENT S)
                    (2 THEME D))
                (C-NAME . "to conduct")
                (EXAMPLE . "katIlar sesi en iyi iletir."))
            (SENSE2
```

```
                (CONST . T)
                (V-CAT PROCESS-ACTION)
                (T-ROLE
                    (1 AGENT S)
                    (2 THEME D))
                (C-NAME . "to convey")
                (EXAMPLE . "yardImI ilettiler."))))
     (ARG-ST2
         ...))
    (ALIAS-LIST ))
```

In the first argument structure, there are subject and direct object. The subject is optional, whereas the object is obligatory, and *nominative* or *accusative* case-marked. These are morphological and syntactic constraints specified in `MORPH` and `SYN` slots of the arguments, and no other constraint is posed by this argument structure. There are two senses associated with this structure. The first poses a semantic constraint in `CONST` slot, which requires that the direct object must be an instance of `POWER-ENERGY-PHYSICALOBJECT` class, like electricity or sound. Then it gives verb category, which is *process-action*, mapping of thematic roles to subcategorization patterns, which maps agent to subject and theme to direct object, and concept name, which is *to conduct*, with an example sentence. The second sense does not pose any additional constraint. The verb category and thematic role mapping of this sense are the same with those of the previous one. Then, the concept name is given as *to convey* with an example sentence.

2.3 Example Work

Due to the growing needs of NLP systems for larger and richer lexicons, the cost of designing and developing lexicons with broad coverage and adequately rich information content is getting high. An example work, which has developed such large lexical resources, may be the Electronic Dictionary Research (EDR) project (Japan, 1990), which run for 9 years, costed 100 million US dollars and intended to develop bilingual resources for English and Japanese containing 200,000 words, term banks containing 100,000 words, and a concept dictionary containing 400,000 concepts. Although the development is aided by special tools, the actual effort is due to the researchers themselves [1].

In order to avoid such high costs, the research institutions and companies are trying to combine their efforts in developing publicly available, large scale language resources, which have adequate information content, and are generic enough (multifunctional) to satisfy various requirements of wide range of NLP applications. Examples of such efforts include ESPRIT BRA (Basic Research Action) ACQUILEX aiming reuse of information extracted from machine-readable dictionaries, WordNet Project at Princeton, which created a large network of word senses related with semantic relations, and LRE EAGLES (Expert Advisory Group on Language Engineering Standards) project, which tries to reach a common lexical specification at some level of linguistic detail among European languages [6].

In the rest of this section, we will concentrate on the EAGLES project. The information given below is mainly received from Monachini and Calzolari [9]. The objective of this work is to propose a common set of morphosyntactic features encoded in lexicons and corpora in European languages, namely Italian, English, German, Dutch, Greek, French, Danish, Spanish, and Portuguese.

The project has gone through three phases:

1. to survey previous work on encoding morphosyntactic phenomena in lexicons and text corpora, e.g., on MULTILEX and GENELEX models, etc.,

2. to work on linguistic annotation of text and lexical description in lexicons to reach a compatible set of features,

3. to test the common proposal by applying concretely to European languages.

The common set of features came after the completion of the second phase, and is described in three main levels corresponding to the level of *obligatoriness*:

1. *Level 0* contains only the part-of-speech category, which is the unique obligatory feature.

2. *Level 1* gives grammatical features, such as gender, number, person, etc. These are generally encoded in lexicons and corpora, and called *recommended features*, which constitute the minimal core set of common features.

3. *Level 2* is subdivided into two:

   - *Level 2a* contains features which are common to languages, but either not generally encoded in lexicons and corpora or not purely morphosyntactic (e.g., countability for nouns). These are considered as *optional features*.

   - *Level 2b* gives *language-specific features*.

The multilayered description, instead of a flat one, gives more flexibility in choosing the level detail in specification to match the requirements of applications. As going down from Level 0 to Level 2, the description reaches finer granularity, and the information encoded increases. Additionally, this type of description helps to extend or update the framework.

The aim of the common proposal is not to pose a complete specification ready to implement, but to pose a basic set of features and to leave the rest to language-specific applications.

The last phase of the project is the testing of the common proposal in a multilingual framework, namely the MULTEXT project. The aim of MULTEXT partners is to design and implement a set of tools for corpus-based research and a corpus in that multilingual framework. The tasks involved are developing a common specification for the MULTEXT lexicon and a tagset for MULTEXT corpus. The partners evaluated the common proposal at Level 1 (recommended features) by also considering language-specific issues. The result is that the common set of features fits well to the description of partners, but needs further language-specific detail.

Chapter 3

A Lexicon Design for Turkish

All natural language processing systems, such as parsers, generators, taggers, need to access a lexicon of the words in the language. The information provided by the lexicon includes:

- morphosyntactic,
- syntactic, and
- semantic information.

In this thesis, we have designed a comprehensive lexicon for Turkish, and integrated it with a morphological processor, so that the overall system is capable of providing the feature structures for all interpretations of an input word form (with multiple senses incorporated).

For instance, consider the input word form *kazma*; first, the morphological processor receives this input, and provides its analysis to the static lexicon. There are three possible interpretations:

1. *kazma* (noun, *pickaxe*),
2. *kaz*+NEG (verb, *don't dig*), and
3. *kaz*+INF (infinitive, *digging*),

for which the static lexicon produces feature structures for all senses of the root words involved. Moreover, the lexicon allows the interfacing system to constraint the output. For example, the final category feature of the root word in the input surface form can be restricted to, say, verb.

In this case, only information about the second interpretation, *don't dig*, will be released by the system. Chapter 4 describes this process in detail.

By separating the system into two parts, that is a morphological analyzer and a static lexicon, we make use of the morphological processor previously implemented and abstract the process of parsing surface forms. Hence, designing a static lexicon and interfacing it with the morphological processor is sufficient to construct a lexicon system.

In this chapter we will present the detailed design of our static lexicon, that is the associated feature structures with each of the lexical categories in Turkish. The procedural aspects (i.e., how feature structures are produced) are described in Chapter 4. We will first introduce the main lexical categories, then describe each one in detail with the associated feature structures.

3.1 Lexicon Architecture

The Figure 3.1 briefly describes the architecture of our lexicon, which consists of a morphological processor, a static lexicon, and a module applying restrictions.

The input to the system is a query form, which consists of two parts: a word form and a set of features placing constraints in the output. The word form is first received and processed by the morphological processor, whose output is the possible interpretations of the word form. Then, the static lexicon attaches features to all senses of the root words of these interpretations, and outputs the feature structures. But before the result is released, the feature structures that do not satisfy the restrictions are eliminated, and the rest is the actual output of the system. The details of this procedure are given in Chapter 4.

3.2 Lexical Representation Langugage

The lexical representation language that we will use in the rest of this chapter is feature structures. A feature structures is a list of <*feature name*:*feature value*> pairs, in which at most one pair with a given feature name can be present. The value of a feature name may be an atom or a feature structure again. Here are some examples of feature structures:[1]

$$\begin{bmatrix} \mathrm{F} & \mathrm{a} \\ \mathrm{G} & \mathrm{b} \end{bmatrix}$$

[1] See Shieber [12] for a detailed description of feature structures.

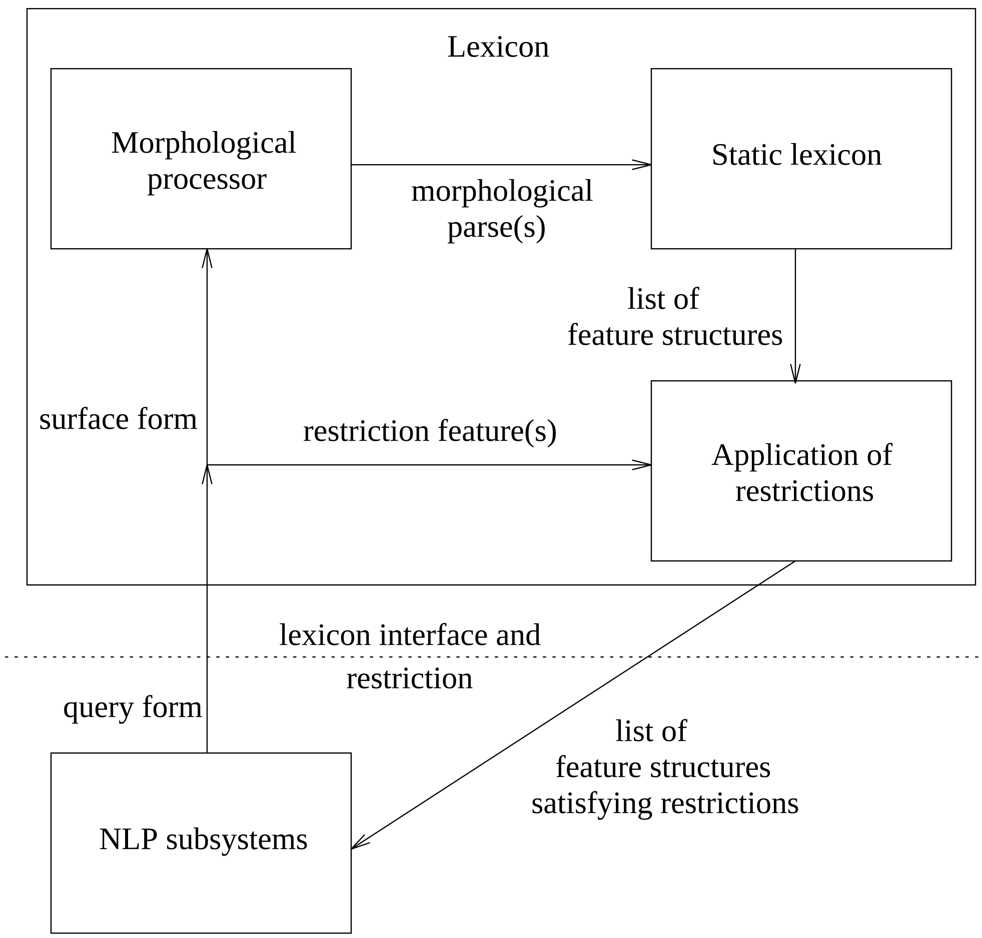

Figure 3.1: Architecture of the lexicon.

$$\begin{bmatrix} \mathrm{F} & \begin{bmatrix} \mathrm{G} & \mathrm{a} \\ \mathrm{H} & \mathrm{b} \end{bmatrix} \\ \mathrm{I} & \mathrm{c} \end{bmatrix}$$

3.3 Lexical Categories

Figure 3.2 shows the main lexical categories of Turkish in our lexicon. All the lexicon categories are depicted in Tables A.1 and A.2 on page 114.

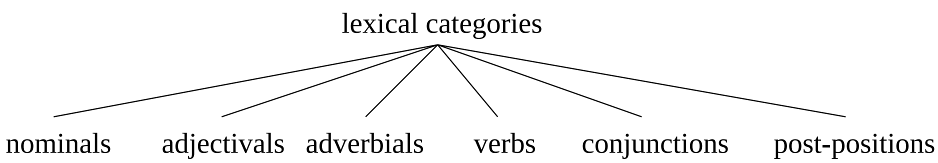

Figure 3.2: The main lexical categories of Turkish.

Each word in the lexicon has the following feature structure:

$${}_{word}\begin{bmatrix} \mathrm{CAT} & \begin{bmatrix} \mathrm{MAJ} & \mathit{maj} & \\ \mathrm{MIN} & \mathit{min} & \text{(default: none)} \\ \mathrm{SUB} & \mathit{sub} & \text{(default: none)} \\ \mathrm{SSUB} & \mathit{ssub} & \text{(default: none)} \\ \mathrm{SSSUB} & \mathit{sssub} & \text{(default: none)} \end{bmatrix} \\ \mathrm{MORPH} & \begin{bmatrix} \mathrm{STEM} & \mathit{stem} \\ \mathrm{FORM} & \text{lexical/derived (default: lexical)} \end{bmatrix} \\ \mathrm{SEM} & \begin{bmatrix} \mathrm{CONCEPT} & \mathit{concept} \end{bmatrix} \\ \mathrm{PHON} & \mathit{phon} \end{bmatrix}$$

Thus, each word has category information in CAT feature as a 5-tuple describing major, minor and subcategories, STEM and FORM as morphosyntactic features, CONCEPT as semantic fetaure, and phonology. The major and minor categories and the concept, which uniquely determine the word with its sense are given in this feature structure. Additionally, the form, which take *lexical* or *derived* values, the stem and the phonology, which is the combination of the stem and inflections are also present in this structure, e.g., *kitap* (*book*) vs. *kitaplarım* (*my books*).

3.4 Nominals

This section describes the representation of nominals in our lexicon. As shown in Figure 3.3, nominals are divided into three subcategories:

- nouns,
- pronouns,
- sentential heads which function as nominals.

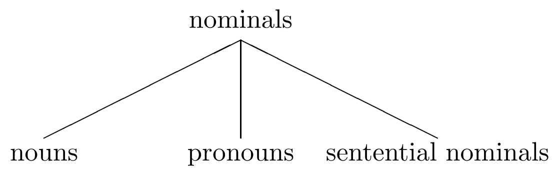

Figure 3.3: Subcategories of nominals.

Figure 3.4 gives the detailed categorization for the nominal category.[2]

maj	*min*	*sub*	*ssub*	*sssub*
nominal	noun	common		
		proper		
	pronoun	personal		
		demonstrative		
		reflexive		
		indefinite		
		quantification		
		question		
	sentential	act	infinitive	ma
				mak
				yış
		fact	participle	dık
				yacak

Figure 3.4: Lexicon categories of nominals.

Each nominal has the following additional features, which represent the inflections of the word:

[2] The three subcategories of infinitives and the two subcategories of participles represent the verbal forms derived using the suffixes -*mA*, -*mAk*, -*yHş*, -*dHk*, and -*yAcAk*. These will be explained later in detail.

The notation for suffixes follows this convention: A and H represent unrounded (i.e., $\{a,\ e\}$) and high vowels (i.e., $\{ı,\ i,\ u,\ ü\}$), respectively. The first y in the suffixes may drop.

$$
{}_{nominal}\left[\begin{array}{ll}
\text{MORPH} & \left[\begin{array}{lll}
\text{CASE} & \textit{case} & \text{(default: none)} \\
\text{AGR} & \textit{agr} & \text{(default: none)} \\
\text{POSS} & \textit{poss} & \text{(default: none)}
\end{array}\right]
\end{array}\right]
$$

A nominal may be case-marked as

- nominative,
- accusative,
- dative,
- locative,
- ablative,
- genitive,
- instrumental,
- equative.

Third person singular and *plural* suffixes are the possible values for the agreement marker of nouns and sentential heads. Pronouns may take *first*, *second*, and *third person singular* and *plural* agreement markers. All three types of nominals may take possessive suffix, which is one of the six person suffixes and *none*.

In the following sections we will describe the subcategories of nominals in detail.

3.4.1 Nouns

Nouns denote the entities in the world, such as objects, events, concepts, etc. As shown in Figure 3.5, nouns can be further divided into two subcategories as *common* and *proper nouns*. These are described in detail in the next two sections.

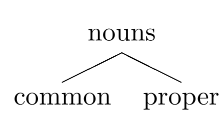

Figure 3.5: Subcategories of nouns.

Common Nouns

Common nouns denote classes of entities. Figure 3.6 depicts the two forms of common nouns: *lexical* and *derived*. Only lexical common nouns are represented in our lexicon as lexical entries, however, the system can produce feature structures for derived forms. For example, computation of the feature structure for *evdekiler* (*those that are at home*) requires the retrieval of the feature structure of the noun *ev* (*home*) and the derivation of it to an adjective (*evdeki* (*that is at home*)) and then to the noun *evdekiler* (see the derivation tree for *evdekiler* in Figure 3.7).

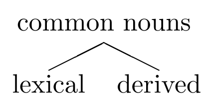

Figure 3.6: Forms of common nouns.

Common nouns have the following additional features: subcategorization and a set of semantic properties such as countability and animateness.

$$
{}_{common}\left[\begin{array}{ll}
\mathrm{SYN} & \left[\begin{array}{ll}\mathrm{SUBCAT} & \left\{\begin{array}{l} constraint_1,\ \ldots, \\ constraint_i,\ \ldots, \\ constraint_n \end{array}\right\} \text{(default: none)}\end{array}\right] \\
\mathrm{SEM} & \left[\begin{array}{ll}
\mathrm{MATERIAL} & +/- \\
\mathrm{UNIT} & +/- \\
\mathrm{CONTAINER} & +/- \\
\mathrm{COUNTABLE} & +/- \\
\mathrm{SPATIAL} & +/- \\
\mathrm{TEMPORAL} & +/- \\
\mathrm{ANIMATE} & +/-
\end{array}\right]
\end{array}\right]
$$

$$
{}_{constraint_i}\left[\begin{array}{ll}
\mathrm{CAT} & \left[\begin{array}{ll}
\mathrm{MAJ} & \text{nominal} \\
\mathrm{MIN} & min \\
\mathrm{SUB} & sub \\
\mathrm{SSUB} & ssub \\
\mathrm{SSSUB} & sssub
\end{array}\right] \\
\mathrm{MORPH} & \left[\begin{array}{ll}
\mathrm{CASE} & case \\
\mathrm{POSS} & poss
\end{array}\right] \\
\mathrm{SEM} & []
\end{array}\right]
$$

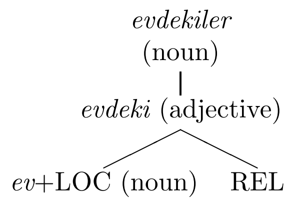

Figure 3.7: Derivation history of *evdekiler*.

The semantic features may only take + or − values. This is on the sense basis, since senses may have different semantic properties; for example, *ekin* (*culture*) is an abstract entity, whereas *ekin* (*crop*) is not. The default value for the semantic features is −.

The subcategorization information consists of a list of constraints on any complement of the common noun. The application of constraints is in *disjunctive* fashion. This concept will be extended to cover more than one complement (e.g., subject, objects, etc.) in Section 3.7, when the *verb* category is introduced. Constraints on the complements of common nouns are of three types: category, case and possessive markings, and semantic properties. Note that the constraint structure for common nouns is simpler than that for verbs. For instance, constraint structure for the current category does not constrain the stem and agreement features of the arguments.

In the next sections we will describe the two forms of common nouns in detail with examples.

Lexical Common Nouns As mentioned above, this form of common nouns are present in the lexicon, and the retrieval does not involve any computation of features. The following are examples of common nouns in lexical form: *kum* (*sand*), *kalem* (*pencil*), *ihtiyaç* (*need*), *sabah* (*morning*), *çarşamba* (*Wednesday*), *ilkbahar* (*spring*), *aşağı* (*bottom*).

As an example, consider the common noun *ihtiyacı* (*his/her/its need*), as used in (3):[3]

(3) a. Utku'nun senin bu işi yapmana
`Utku+GEN you+GEN this job+ACC do+INF+P2SG`
ihtiyacı var.
`need+P3SG existent+PRES+3SG`
'Utku needs you to do this job.'

b. Bunun için sana/Bilge'ye ihtiyacımız var.
`this+GEN for you/Bilge+DAT need+P1PL existent+PRES+3SG`
'We need you/Bilge for this.'

[3] Note that some of the features are not shown; they take the default values specified.

$$
\textit{lexical common}\begin{bmatrix}
\mathrm{CAT} & \begin{bmatrix} \mathrm{MAJ} & \mathrm{nominal} \\ \mathrm{MIN} & \mathrm{noun} \\ \mathrm{SUB} & \mathrm{common} \end{bmatrix} \\
\mathrm{MORPH} & \begin{bmatrix} \mathrm{STEM} & \text{“ihtiyaç”} \\ \mathrm{FORM} & \mathrm{lexical} \\ \mathrm{CASE} & \mathrm{nom} \\ \mathrm{AGR} & \mathrm{3sg} \\ \mathrm{POSS} & \mathrm{3sg} \end{bmatrix} \\
\mathrm{SYN} & \begin{bmatrix} \mathrm{SUBCAT} & \left\{ constraint_1, constraint_2 \right\} \end{bmatrix} \\
\mathrm{SEM} & \begin{bmatrix} \mathrm{CONCEPT} & \text{\#ihtiyaç-(need)} \end{bmatrix} \\
\mathrm{PHON} & \text{“ihtiyaç”}
\end{bmatrix}
$$

$$
constraint_1\begin{bmatrix}
\mathrm{CAT} & \begin{bmatrix} \mathrm{MAJ} & \mathrm{nominal} \\ \mathrm{MIN} & \left\{ \text{noun, pronoun} \right\} \end{bmatrix} \\
\mathrm{MORPH} & \begin{bmatrix} \mathrm{CASE} & \mathrm{dat} \end{bmatrix}
\end{bmatrix}
$$

$$
constraint_2\begin{bmatrix}
\mathrm{CAT} & \begin{bmatrix} \mathrm{MAJ} & \mathrm{nominal} \\ \mathrm{MIN} & \mathrm{sentential} \\ \mathrm{SUB} & \mathrm{act} \\ \mathrm{SSUB} & \mathrm{infinitive} \\ \mathrm{SSSUB} & \mathrm{ma} \end{bmatrix} \\
\mathrm{MORPH} & \begin{bmatrix} \mathrm{CASE} & \mathrm{dat} \end{bmatrix}
\end{bmatrix}
$$

The feature structure of *ihtiyacı* contains information stating that *ihtiyacı* is a common noun in lexical form, inflected from *ihtiyaç* with *3sg* agreement and possessive markers. It also specifies that the complement of *ihtiyacı* should be case-marked as *dative* and may be in one the two forms: noun or pronoun, and infinitive derived with the suffix *-mA*. Example sentences in (3) depict these usages.

The following is another example, the common noun *geceye* (*to the night*), as used in (4):

(4) Dün geceye kadar oraya gitmek
`yesterday night+DAT until there+DAT go+INF`
konusunda karar vermiş değildim.
`topic+P3SG+LOC decide+NARR NOT+PAST+1SG`
'I had not decided on going there until last night.'

$$
\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“gece”} \\ \text{FORM} & \text{lexical} \\ \text{CASE} & \text{dat} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \text{SUBCAT} & \text{none} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#gece-(night)} \\ \text{COUNTABLE} & + \\ \text{TEMPORAL} & + \end{bmatrix} \\
\text{PHON} & \text{“geceye”}
\end{bmatrix}_{\textit{lexical common}}
$$

The feature structure above gives the following information: *geceye* is a common noun in lexical form, inflected from the common noun *gece* with *3sg* agreement and *dative* case markers. It is countable and states temporality.

Derived Common Nouns Derived forms of common nouns are not represented directly in the lexicon. However, in order to produce feature structures, the lexicon employs the derivation information provided by the morphological processor. This information mainly consists of the target category and the derivational suffixes. The rest of the information (such as argument structure, thematic roles, concept, and stem) are supplied by the lexicon. The details of this process are described in Chapter 4.

Each derived common noun has the following additional features:

$$
\begin{bmatrix}
\text{MORPH} & \begin{bmatrix} \text{DERV-SUFFIX} & \textit{derv-suffix} \text{ (default: none)} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{ROLES} & \textit{roles} \text{ (default: none)} \end{bmatrix}
\end{bmatrix}_{\textit{derived common}}
$$

These give the suffix used in the derivation and the semantic functions involved. The latter stores the thematic roles of the lexical verb which is involved somewhere in the derivation process. For example, the derived common noun *yazıcı* (*writer*) has the thematic roles of the verb *yaz-* (*write*), since the derivation process carries the thematic role information through categories. The type of this feature's value is given in Section 3.7.

The derivation suffix may take one of the following values: *-cH*, *-cHk*, *-lHk*, *-yHcH*, *-mAzlHk*, *-yAmAzHk*, *-mAcA*, *-yAsH* and *none*.

However, there is the problem of predicting the semantic properties of derived common nouns, and this is not an easy task. For example, consider *akşamcı* (*heavy drinker*) and *öğlenci* (*the student attending the afternoon session of a school*), which are both derived from common nouns with the suffix *-cH*. The semantics is, however, rather unpredictable. The current system does not attempt to predict those values. Instead, the default values are used; but these may not necessarily be the correct values for the word in consideration. Prediction of these values is beyond the scope of our work.

There are four types of derivation to derived common nouns:

- Nominal derivation: This type of derivation uses the suffixes *-cH*, *-cHk*, *-lHk*, as in the examples *kapıcı* (*doorkeeper*), *kitapçık* (*booklet*), and *kitaplık* (*bookcase*).

  Consider the feature structure for the common noun *tamircim* (*my repairman*), as used in the example sentence below:

(5)

Her zaman	olduğu	gibi,	tamircim
`always`	`happen+PART+P3SG`	`like`	`repairman+P1SG`

işini	çok	iyi	yaptı.
`job+P2SG`	`very`	`well`	`do+PAST+3SG`

'As it is always the case, my repairman did his job very well.'

$$
\mathit{derived\ common}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \boxed{1}\\ \text{FORM} & \text{derived}\\ \text{CASE} & \text{nom}\\ \text{AGR} & \text{3sg}\\ \text{POSS} & \text{1sg}\\ \text{DERV-SUFFIX} & \text{``cı''}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\ \text{none}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{f}_{\text{cı}}(\boxed{3})\end{bmatrix}\\
\text{PHON} & \text{``tamircim''}
\end{bmatrix}
$$

$$\boxed{1}\ {}_{\textit{lexical common}}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix} \\ \text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“tamir”}\\ \text{FORM} & \text{lexical}\end{bmatrix} \\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\ \text{none}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{3}\ \text{\#tamir-(repair)}\end{bmatrix} \\ \text{PHON} & \text{“tamir”} \end{bmatrix}$$

The feature structure for the noun *tamircim* is produced first retrieving the features of *tamir* (*repair*) and filling a template for derived common nouns appropriately. Some of the feature values are obtained from the features of *tamir* (e.g., subcategorization information), some of them are supplied by the morphological processor (e.g., inflectional and derivational suffixes), and the rest is provided by the static lexicon.

The feature structure above gives the following information: the word *tamircim* is a common noun derived from *tamir* with the suffix *cH*, and inflected with *3sg* and *1sg* agreement and possessive markers, respectively. *Tamircim* does not have subcategorization information. It also includes all the features of *tamir*.

- Adjectival derivation: Derivation from adjectival uses the suffix *-lHk*, e.g., *iyilik* (*goodness*), *temizlik* (*cleanliness*). But, derivation without suffix is also possible as in the following examples, though this is not productive:

(6)	– borçlu	‘that owing debt’,
	– akıllı	‘intelligent’,
	– geridekine	‘to the one behind’.

This is also possible in the case of participles (compare with participles in Section 3.4.3), such as

(7)	– getirdiğimi	‘the thing that I brought’,
	– gelene	‘to the one that came/coming’.

As described in the section on qualitative adjectives, this type of adjectivals are derived from verbs, and by dropping the head of the phrase that they modify and taking their inflectional suffixes, they become nominals. An example is given in (8):

(8)	a.	Buraya	*gelen*	*adamı*	gördün	mü?
		`here+DAT`	`come+PART`	`man+ACC`	`see+PAST+2SG`	`QUES`
		‘Did you see the man that came here?’				

b. Buraya *geleni* gördün mü?
`here+DAT come+PART+ACC see+PAST+2SG QUES`
'Did you see the one that came here?'

In sentence (8a), the verbal form of gapped relative clause, *buraya gelen*, acting as the modifier of *adam* (*man*) takes the inflections of *adam*, and functions as a nominal.

There are two types of participles (see Underhill [15]):

– *subject* (such as *gelen adam* (*the man that came/is coming*)),
– *object* (such as *getirdiğim kitap* (*the book that I brought*)).

In order for an object participle to be used as a nominal (specifically common noun), the verb from which the adjectival is derived should take a direct object. Otherwise, the nominal represents a fact. For example, the verb, *gel-* (*come*), may not take a direct object argument, thus the nominal, *geldiğini* in (9a) represents a fact. In (9b), however, the nominal, *getirdiğini*, has two readings: a fact and a derived common noun.

(9) a. Taner'in geldiğini biliyorum.
`Taner+GEN come+PART+P3SG know+PROG+1SG`
'I know that Taner came.'

b. Taner'in getirdiğini biliyorum.
`Taner+GEN bring+PART+P3SG know+PROG+1SG`
'I know that Taner brought something.'
'I know the thing that Taner brought.'

- Verb derivation: This derivation type uses the suffixes *-yHcH*, *-mAcA*, *-mAzlHk*, *-yAmAzlHk*, and *-yAsH*, as used in the following example nouns: *yazıcı* (*writer*), *koşucu* (*runner*), *koşuşturmaca* (*rush/hurry*), *çekememezlik* (*envy*), *kahrolası* (*damnable*).

- Post-position derivation: Derivation from post-positions do not use any suffix, e.g., *azını* (*the one that is little*), *yukarısına* (*to the one that is above*).

Proper nouns

Proper nouns are used to refer to unique entities in the world. The only additional feature that proper nouns have states that they are always *definite*, as in the examples *Kurtuluş*, *Kemal*, *Oflazer*, *Bilkent*, and *Ankara*.

$$_{proper}\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{DEFINITE} & +\end{bmatrix}\end{bmatrix}$$

As used in (10), the following is the feature structure of the proper noun *Kurtuluş*:

(10) Kurtuluş yarım saat içinde burada olacak.
`Kurtuluş half hour in here+LOC be+FUT+3SG`
'Kurtuluş will be here in half an hour.'

$$
{}_{proper}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{proper} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“Kurtuluş”} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#Kurtuluş-(Kurtuluş)} \\ \text{DEFINITE} & + \end{bmatrix} \\
\text{PHON} & \text{“Kurtuluş”}
\end{bmatrix}
$$

3.4.2 Pronouns

Pronouns are used in place of nouns in sentences, phrases, etc. (see Ediskun [3] and Koç [8])and subdivided into six categories, as shown in Figure 3.8.

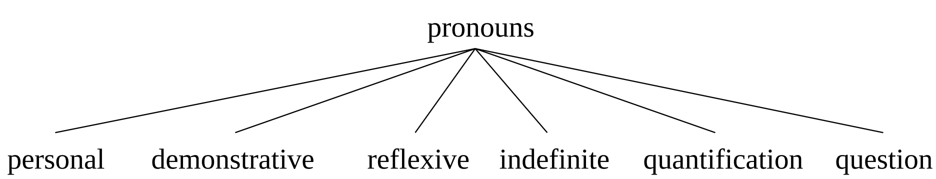

Figure 3.8: Subcategories of pronouns.

Each pronoun also has the following semantic feature, which takes + value for personal, reflexive and demonstrative pronouns, and − value for the other subcategories.

$$
{}_{pronoun}\begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{DEFINITE} & +/- \text{ (default: } -) \end{bmatrix} \end{bmatrix}
$$

In the following sections we will give examples for each subcategory of pronouns.

Personal pronouns

Personal pronouns are used to denote the speaker, the one spoken to, and the one spoken of. This category consists of pronouns *ben* (*I*), *sen* (*you*), *o* (*he/she/it*), *biz/bizler* (*we*), *siz/sizler* (*you*), and *onlar* (*they*). Personal pronouns may take all of the six person suffixes as the agreement marker, but may not take a possessive marker.

Demonstrative pronouns

Demonstrative pronouns denote the entities by showing them, but without mentioning their actual names. The following are examples of demonstrative pronouns: *bu* (*this*), *şu* (*that*), *bunlar* (*these*). Like personal pronouns, this category of pronouns does not take a possessive marker. *3sg* and *3pl* suffixes are the possible values for the agreement marker. The following is the feature structure of *onlar* (*they*), as used in (11):

(11)	Bunu	yapanın	onlar	olduğundan	eminim.
	`this+ACC`	`do+PART+GEN`	`they`	`be+PART+P3SG+ABL`	`sure+PRES+1SG`
	‘They, I am sure, did this.’				

$$\left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{pronoun} \\ \text{SUB} & \text{demonstrative} \end{array}\right] \\ \text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \text{“o”} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3pl} \\ \text{POSS} & \text{none} \end{array}\right] \\ \text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \text{\#o-(he/she/it)} \\ \text{DEFINITE} & + \end{array}\right] \\ \text{PHON} & \text{“onlar”} \end{array}\right]_{\textit{demonstrative pronoun}}$$

Reflexive pronouns

Reflexive pronouns are words denoting the person or the thing on which the action in the sentence has an effect. This category consists of the pronouns *kendim* (*myself*), *kendin* (*yourself*), *kendi/kendisi* (*herself/himself/itself*), *kendimiz* (*ourselves*), *kendiniz* (*yourselves*), and *kendileri* (*themselves*). The agreement and possessive markers take the same value, which is one of the six person suffixes, e.g., it is *3pl* suffix for *kendileri*. The same holds true for the indefinite and quantification pronouns.

Indefinite pronouns

Indefinite and quantification pronouns denote entities without showing them explicitly. The difference between the two is that quantification pronouns recall the existence of more than one entity. All indefinite pronouns are inflected forms of the root word *biri* and *kimi*, e.g., *biri/birisi* (*someone*), *birimiz* (*one of us*), *kiminiz* (*some of you*), *kimileri* (*some of them*).[4]

Quantification pronouns

There are two forms of quantification pronouns: *lexical* and *derived*.

Lexical The following are examples of quantification pronouns in lexical form: *kimisi* (*some of them*), *kimimiz* (*some of us*), *bazısı* (*some of them*), *birçoğu* (*most of them*), *çoğumuz* (*most of us*), *herbirimiz* (*each of us*), *tümümüz* (*all of us*), *hepsi* (*all of them*).

Consider the feature structure of the quantification pronoun *birçoğu* (*most of them*), as used in (12):

(12)	Kötü	hava	koşulları	yüzünden,	öğrencilerin
	`bad`	`weather`	`condition+3PL+P3SG`	`due to`	`student+3PL+GEN`
	birçoğu		gelemedi.		
	`most of them`		`come+NEG+PAST+3SG`		

‘Due to bad weather conditions, most of the students couldn’t come.’

$$
\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{pronoun} \\ \text{SUB} & \text{quantification} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“birçok”} \\ \text{FORM} & \text{lexical} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3pl} \\ \text{POSS} & \text{3pl} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#birçok-(most of \ldots)} \end{bmatrix} \\
\text{PHON} & \text{“birçoğu”}
\end{bmatrix}_{\textit{lexical quantification pronoun}}
$$

[4] Note that the inflected forms of *iki*, *üç*, etc. (such as *ikiniz* (*two of you*)) are classified as quantification pronouns. However, this is not productive.

Derived The derivation to quantification pronouns is possible only from quantification adjectives, e.g., *ikisi* (*two of them*), *üçünüz* (*you three*). The derivation process is not productive: for example, **ikileri* is not a quantification pronoun. The derivation does not use a suffix.

Each derived quantification pronoun has the following additional feature:

$$_{\textit{derived quantification pronoun}}\begin{bmatrix}\text{MORPH} & \begin{bmatrix}\text{DERV-SUFFIX} & \textit{none}\end{bmatrix}\end{bmatrix}$$

Question pronouns

This category of pronouns look for entities by asking questions. The following are examples of question pronouns: *kim/kimler* (*who*), *ne* (*what*), *hangisi* (*which of them*), *hanginiz* (*which of you*). For the agreement and possessive markers, there are two cases:

- they both take the same value, which is one of the six person suffixes, e.g., it is *2pl* for *hanginiz*,
- agreement marker takes one of *3sg* and *3pl* suffixes, and possessive marker does not take any value, e.g., *kim* vs. *kimler*.

3.4.3 Sentential Nominals

In this section we will describe sentential nominals, which head sentences and function as nominals in syntax. As shown in Figure 3.9, sentential nominals are divided into two subcategories: *acts* and *facts*.

sentential nominals

acts facts

Figure 3.9: Subcategories of sentential nominals.

Each sentential nominal has the following additional features:

$$_{\textit{sentential}}\begin{bmatrix}\text{MORPH} & \begin{bmatrix}\text{DERV-SUFFIX} & \textit{derv-suffix}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \textit{subcat}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{ROLES} & \textit{roles}\end{bmatrix}\end{bmatrix}$$

The DERV-SUFFIX feature takes one of the following: *-mAk*, *-mA*, *-yHş*, *-dHk*, and *-yAcAk*. Subcategorization information and thematic roles are also present in this feature structure.

Acts

The only subcategory of acts is *infinitives*, which is described next.

Infinitives Infinitives may be further divided into three subcategories, which are derived from verbs with the suffixes *-mA*, *-mAk*, and *-yHş*, respectively, as shown in Figure 3.10. The derivation with *-mAk* is indefinite, i.e., the infinitive does not take a possessive marker, while the other two may or may not take this inflection.

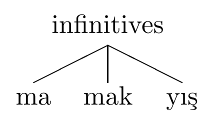

Figure 3.10: Subcategories of infinitives.

The following are examples of infinitives: *gelmesi* (*his coming*), *gelişi* (*his coming*), *koşmak* (*to run*), *çalışmaktan* (*from working*). As an example, consider the following feature structure for the infinitive *bilmek* (*to know*), as used in (13):[5]

(13) a. Tolga'nin dün buraya neden geldiğini
`Tolga+GEN yesterday here+DAT why come+PART+P3SG+ACC`
bilmek sana birşey kazandırmaz.
`to know you+DAT something gain+CAUS+NEG+ARST+3SG`
'You will not gain anything by knowing why Tolga came here yesterday.'

b. Araba kullanmayı biliyor musun?
`car drive+INF+ACC know+PRES QUES+2SG`
'Do you know how to drive?'

c. Bu işi nasıl bitireceğimi biliyorum.
`this job+ACC how end+PART+P1SG+ACC know+PRES+1SG`
'I know how to end this thing.'

[5] Sentences (13b) and (13c) are given to examplify the argument structure of the verb *bil-*.

$$
{}_{mak}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{mak} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \boxed{1} \\ \text{FORM} & \text{derived} \\ \text{DERV-SUFFIX} & \text{``mak''} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \end{array}\right] \\
\text{SYN} & \left[\begin{array}{ll} \text{SUBCAT} & \boxed{2} \end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \text{f}_{mak}(\boxed{4}) \\ \text{ROLES} & \boxed{3} \end{array}\right] \\
\text{PHON} & \text{``bilmek''}
\end{array}\right]
$$

$$
\boxed{1}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predicative}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“bil”}\\ \text{FORM} & \text{lexical}\\ \text{SENSE} & \text{pos}\end{array}\right]\\
\text{SYN} & \left[\text{SUBCAT}\ \boxed{2}\left\langle \begin{array}{l}
\boxed{5}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{subject}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\mathit{constraint}_1\right\}\end{array}\right],\\
\boxed{6}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{dir-obj}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\begin{array}{l}\mathit{constraint}_2,\\ \mathit{constraint}_3,\\ \mathit{constraint}_4,\\ \mathit{constraint}_5\end{array}\right\}\end{array}\right]
\end{array}\right\rangle\right]\\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & \boxed{4}\ \text{\#bil-(to know)}\\ \text{ROLES} & \boxed{3}\left[\begin{array}{ll}\text{AGENT} & \boxed{5}\\ \text{THEME} & \boxed{6}\end{array}\right]\end{array}\right]\\
\text{PHON} & \text{“bil”}
\end{array}\right]_{\textit{lexical predicative verb}}
$$

$$
\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \left\{\text{noun, pronoun}\right\}\end{array}\right]\\
\text{MORPH} & \left[\text{CASE}\ \ \text{nom}\right]
\end{array}\right]_{\mathit{constraint}_1}
$$

$$
\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\end{array}\right]\\
\text{MORPH} & \left[\text{CASE}\ \ \left\{\text{acc, nom}\right\}\right]
\end{array}\right]_{\mathit{constraint}_2}
$$

$$
\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{pronoun}\end{array}\right]\\
\text{MORPH} & \left[\text{CASE}\ \ \text{acc}\right]
\end{array}\right]_{\mathit{constraint}_3}
$$

$$constraint_4 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{ma} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{acc} \end{bmatrix} \end{bmatrix}$$

$$constraint_5 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{fact} \\ \text{SSUB} & \text{participle} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{acc} \\ \text{POSS} & \neg\text{none} \end{bmatrix} \end{bmatrix}$$

Facts

The only subcategory of facts is *participles*, which is described next.

Participles Participles may be further divided into two subcategories, which are derived from verbs with the suffixes *-dHk* and *-yAcAk*, respectively, as shown in Figure 3.11. Both subcategories take possessive markings.

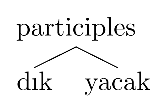

Figure 3.11: Subcategories of participles.

The following are two examples of participles describing facts:

(14)	– geldiği	'the fact that he came',
	– geleceğini	'the fact that he is going to come'.

Note that Section 3.4.1 describes the participles functioning as common nouns. As an example of participles acting as sentential nominals and common nouns, consider (15a), which contains a sentence with two parses. The first mentions about *the thing that Gamze brought*, and the participle, *getirdiğini*, used as a common noun. The latter is about *the event that Gamze brought something*, and the participle is used to represent this fact. However, the participle in (15b) can only be used to describe a fact.

(15) a. Gamze'nin Ankara'dan getirdiğini gördüm.
`Gamze+GEN Ankara+ABL bring+PART+P3SG+ACC see+PAST+1SG`
'I saw the thing that Gamze brought from Ankara.'
'I saw that Gamze has brought it from Ankara.'

b. Gamze'nin geldiğini gördüm.
`Gamze+GEN come+PART+P3SG+ACC see+PAST+1SG`
'I saw that Gamze came.'

3.5 Adjectivals

This section describes the representation of adjectivals in our lexicon. Adjectivals are words that describe the properties of nominals (specifically common nouns) in a number of ways, e.g., quality, quantity, etc. and specify them by differentiating from the others. As shown in Figure 3.12, adjectivals consists of two subcategories: *determiners* and *adjectives*. Figure 3.13 shows the hierarchy under the adjectival category.

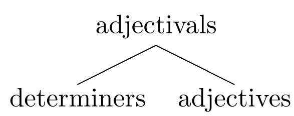

Figure 3.12: Subcategories of adjectivals.

maj	*min*	*sub*	*ssub*
adjectival	determiner	article	
		demonstrative	
		quantifier	
	adjective	quantitative	cardinal
			ordinal
			fraction
			distributive
		qualitative	

Figure 3.13: Lexicon categories of adjectivals.

Each adjectival has the following additional feature structure, which contains syntactic and semantic information. SYN | MODIFIES specifies constraints on the modified of the adjectival including its category, agreement marking and countability. For example, the cardinal adjective *bir* accepts only singular countable common nouns, e.g., *bir kalem* vs. **bir kalemler*.[6]

$$
{}_{\textit{adjectival}}\left[\begin{array}{ll}
\mathrm{SYN} & \left[\begin{array}{ll}
\mathrm{MODIFIES} & \left[\begin{array}{ll}
\mathrm{CAT} & \left[\begin{array}{ll} \mathrm{MAJ} & \text{nominal} \\ \mathrm{MIN} & \text{noun} \\ \mathrm{SUB} & \text{common} \end{array}\right] \\
\mathrm{MORPH} & \left[\begin{array}{ll} \mathrm{AGR} & \textit{agr} \end{array}\right] \\
\mathrm{SEM} & \left[\begin{array}{ll} \mathrm{COUNTABLE} & +/- \end{array}\right]
\end{array}\right]
\end{array}\right] \\
\mathrm{SEM} & \left[\begin{array}{ll} \mathrm{GRADABLE} & +/-/\text{semi (default: } -) \\ \mathrm{QUESTIONAL} & +/- \text{ (default: } -) \end{array}\right]
\end{array}\right]
$$

There are two semantic features. The first one describes the gradability of the adjectival in consideration, e.g., the article *bir* is not gradable, whereas, the adjective *büyük* is. The other one is used to describe whether the adjectival is in questional form, e.g., the following adjectivals are in this form: *kaç* (*how many*), *kaçıncı* (*in what order*), *nasıl* (*how*), *hangi* (*which*).

In the next sections we will describe the subcategories of adjectivals in detail.

3.5.1 Determiners

Determiners are limiting adjectivals: they specify entities by showing them explicitly or indefinitely. As shown in Figure 3.14, determiners are subdivided into three categories: *indefinite article*, *demonstratives* and *quantifiers*, which are described in the next sections.

Indefinite Article

The only article in Turkish is *bir*, as used in (17). As the name implies, this article, like quantifiers, does not show entities explicitly. The feature structure of this article is given below:

[6] The category information states that adjectivals can only modify common nouns, which is not accurate, in fact. Consider the following example:

(16) a.

Ankara'ya	*bu*	*gidişimde*	onunla	konuşacağım.
`Ankara+DAT`	`this`	`go+INF+P2SG+LOC`	`him+DAT`	`talk+FUT+1SG`

'I will talk with him in my next visit to Ankara.'

In this sentence, the demonstrative *bu* modifies a sentential nominal. However, we will omit these and simplify the pattern of modified constituent of adjectival phrases.

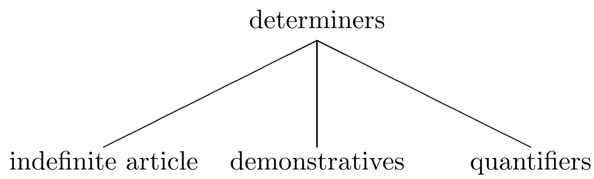

Figure 3.14: Subcategories of determiners.

(17) a. Dilek evinde büyük bir balık besliyor.
`Dilek home+P3SG+LOC big a fish look after+PROG+3SG`
'Dilek is looking after a big fish at her home.'

$$
{}_{article}\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{adjectival} \\ \text{MIN} & \text{determiner} \\ \text{SUB} & \text{article} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“bir”} \end{bmatrix} \\ \text{SYN} & \begin{bmatrix} \text{MODIFIES} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{AGR} & \text{3sg} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{COUNTABLE} & + \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{CONCEPT} & \#\text{bir-(a)} \end{bmatrix} \\ \text{PHON} & \text{“bir”} \end{bmatrix}
$$

Demonstratives

Demonstratives specify entities by showing them explicitly. *Bu* (*this*), *şu* (*that*), *hangi* (*which*) and *diğer* (*other*) are examples of demonstratives. As a specific example, consider *bu* (*this*), which is used in (18):

(18) Bulduğum bu örnek cümle çok saçma.
`devise+PART+P1SG this example sentence very foolish+PRES+3SG`
'This example sentence I devised is foolish.'

$$
{}_{demonstrative}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{determiner}\\ \text{SUB} & \text{demonstrative}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“bu”}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{MODIFIES} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#bu-(this)}\end{bmatrix}\\
\text{PHON} & \text{“bu”}
\end{bmatrix}
$$

Quantifiers

Her (*each*), *bazı/kimi* (*some*), *biraz* (*a little*), *birçok* (*many*), and *bütün* (*all*) are examples of quantifiers. The following is the feature structure of *biraz* (*a little*), as used in the example sentence below:

(19)	Timuçin,	bana	biraz	su	getirir	misin?
	`Timuçin`	`me+DAT`	`a little`	`water`	`bring+ARST`	`QUES+2SG`

‘Timuçin, could you bring me a little water?’

$$
{}_{quantifier}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{determiner}\\ \text{SUB} & \text{quantifier}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“biraz”}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{MODIFIES} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{AGR} & \text{3sg}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{COUNTABLE} & -\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#biraz-(a little)}\end{bmatrix}\\
\text{PHON} & \text{“biraz”}
\end{bmatrix}
$$

3.5.2 Adjectives

Adjectives are used to describe the quantity and quality of entities. Figure 3.15 presents the subcategories of adjectives, which consists of *quantitative* and *qualitative adjectives*. These subcategories are described in the following sections.

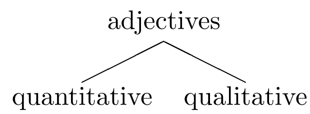

Figure 3.15: Subcategories of adjectives.

Quantitative Adjectives

Quantitative adjectives describe the amount of the entities. This category is further divided into four subcategories, as shown in Figure 3.16.

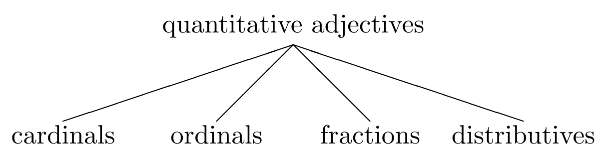

Figure 3.16: Subcategories of quantitative adjectives.

Cardinals Cardinals specify how many of entities are present. The following are examples of cardinals: *bir* (*one*), *iki* (*two*), *yüzlerce* (*hundreds of*), *kaç* (*how many*).

Ordinals Ordinals specify the rank of an entity. The following are examples of ordinals: *birinci/ilk* (*first*), *ikinci* (*second*), *sonuncu* (*last*), *kaçıncı* (*in what order*).

Fractions This category of quantitative adjectives specify the relative size of the parts of an entity. The following are examples of fractions: *bütün/var/tam/tüm* (*whole*), *yarım* (*half*), çeyrek (*one fourth*). The following example demonstrates the fraction adjective usage of *var*, which may not be evident at the first glance:

(20)	Kazanmak	için	var	gücümle	çalıştım.
	`win+INF`	`for`	`whole`	`power+P1SG+INS`	`work+PAST+1SG`
	'I word so hard to win.'				

Distributives *Birer* (*one each*) is an example of distributives, which gives the size of each group that is obtained by dividing an entity into parts equally.

Qualitative Adjectives

Qualitative adjectives describe the properties of the entities. There are two forms of qualitative adjectives: *lexical* and *derived*. In the next sections we will describe these forms in detail with examples.

Each qualitative adjective has the following additional feature, which gives the subcategorization information:

$$_{qualitative\ adj}\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \textit{subcat}\ \text{(default: none)}\end{bmatrix}\end{bmatrix}$$

Lexical The feature structures of this form of adjectives are directly accessible in the lexicon, i.e., no derivation process is involved. The subcategorization information for this form consists of a list of constraints on the only (if any) complement of the adjective (see the example below). The following are examples of qualitative adjectives in lexical form: *memnun* (*pleased*), *iyi* (*good*), *zeki* (*clever*), *küçük* (*small*), *aynı* (*same*), *ertesi* (*next*), *çok* (*many/much*), *sarı* (*yellow*), *nasıl* (*how*).

Consider the feature structure for *memnun* (*pleased*), as used in (22):[7]

(22) a. Ondan memnun bir tek çalışan yok burada.
`him+ABL pleased one unique worker nonexistent+PRES+3SG here+LOC`
'There is no one worker who is pleased from him.'

b. Olayın bu şekilde gelişmesinden memnun
`event+GEN this way+LOC develop+INF+P3SG+ABL pleased`
değiliz.
`NOT+PRES+1SG`
'We are not pleased from the way it develops.'

[7] Note that the argument structure of *memnun*, when used with the auxilary verb *ol-*, is different from that of the adjective usage. *Memnun ol-* (*be happy/satisfied*) is considered as a separate compound verb (see Section 3.7).

(21)
Buna mennun oldum.
`this+DAT be happy+PAST+1SG`
'I am happy with it.'

$$
{}_{\textit{lexical qualitative adj}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{adjective}\\ \text{SUB} & \text{qualitative}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“memnun”}\\ \text{FORM} & \text{lexical}\end{array}\right]\\
\text{SYN} & \left[\begin{array}{ll}\text{MODIFIES} & \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{array}\right]\end{array}\right]\\
\text{SUBCAT} & \left\{\begin{array}{l} constraint_1, constraint_2,\\ constraint_3, constraint_4\end{array}\right\}\end{array}\right]\\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & \text{\#memnun-(pleased)}\\ \text{GRADABLE} & +\end{array}\right]\\
\text{PHON} & \text{“memnun”}
\end{array}\right]
$$

$$
{}_{constraint_1}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \left\{\text{noun, pronoun}\right\}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{CASE} & \text{abl}\end{array}\right]
\end{array}\right]
$$

$$
{}_{constraint_2}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{sentential}\\ \text{SUB} & \text{act}\\ \text{SSUB} & \text{infinitive}\\ \text{SSSUB} & \text{mak}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{CASE} & \text{abl}\\ \text{POSS} & \text{none}\end{array}\right]
\end{array}\right]
$$

$$
{}_{constraint_3}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{sentential}\\ \text{SUB} & \text{act}\\ \text{SSUB} & \text{infinitive}\\ \text{SSSUB} & \text{ma}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{CASE} & \text{abl}\\ \text{POSS} & \neg\text{none}\end{array}\right]
\end{array}\right]
$$

$$
{}_{constraint_4}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{yış} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{CASE} & \text{abl} \end{bmatrix}
\end{bmatrix}
$$

Derived Similar to other categories in derived form, producing feature structures for derived qualitative adjectives requires computation of features.

Each derived qualitative adjective has the following additional features:

$$
{}_{\textit{derived qualitative adj}}\begin{bmatrix}
\text{MORPH} & \begin{bmatrix} \text{DERV-SUFFIX} & \textit{derv-suffix} \\ \text{POSS} & \textit{poss}\ \text{(default: none)} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{ROLES} & \textit{roles}\ \text{(default: none)} \end{bmatrix}
\end{bmatrix}
$$

The derivation suffix may take one of the following values: *-lHk*, *-lH*, *-ki*, *-sHz*, *-sH*, *-yHcH*, *-yAn*, *-yAcAk*, *-dHk*, *-yAsH*, and *none*. The feature MORPH | POSS is used to hold the possessive marking of adjective derived from verb, as in *bildiğim yemek* (*bil*+*dHk*+P1SG *yemek*, *dish that I know*). Possible values for this feature are the six person suffixes. The last feature gives the semantic roles of the verb which is involved in the derivation process.

During the derivation process, since predicting the gradability of the qualitative adjective is difficult, its default value (i.e., it is $-$) is used. For example, adjective *akılsız* (*stupid*) is gradable, while *kolsuz* (*without arm*) is not, that is *çok akılsız* (*very stupid*) vs. **çok kolsuz*. However, the following prediction about the constraints on the complements of the derived qualitative adjectives is generally correct: qualitative adjectives are generally modifiers of common nouns and do not constrain the agreement and countability features of the modified.

There are two possible derivations to qualitative adjectives:

- Nominal derivation: This derivation uses suffixes *-lHk*, *-lH*, *-ki*, *-sHz*, *-sH*, as in *akıllı* (*intelligent*), *evdeki* (*that is at home*), and *çocuksu* (*childish*).

  Consider the feature structure for the derived qualitative adjective, *akıllı* (*intelligent*), as used in the following sentence:

(23)	Akıllı	insanlar	böyle	şeyler	yapmazlar.
	`inteligent`	`people`	`such`	`thing+3PL`	`do+NEG+ARST+3PL`
	‘Intelligent people don’t do this kind of things.’				

$$
{}_{\textit{derived qualitative adj}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{adjective}\\ \text{SUB} & \text{qualitative}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \boxed{1}\\ \text{FORM} & \text{derived}\\ \text{DERV-SUFFIX} & \text{“lı”}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\ \text{none}\\ \text{MODIFIES} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{f}_{\text{lı}}(\boxed{3})\\ \text{ROLES} & \text{none}\end{bmatrix}\\
\text{PHON} & \text{“akıllı”}
\end{bmatrix}
$$

$$
\boxed{1}\ {}_{\textit{lexical common}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{FORM} & \text{lexical}\\ \text{STEM} & \text{“akıl”}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\ \text{none}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{3}\ \text{\#akıl-(intelligence)}\end{bmatrix}\\
\text{PHON} & \text{“akıl”}
\end{bmatrix}
$$

- Verb derivation: This form of derivation uses the following suffixes: *-yHcH*, *-yAn*, *-yAcAk*, *-dHk*, *-yAsH*, and *none*. Verbal form that take suffixes *-yAn*, *-yAcAk*, *-dHk*, and *-yAsH* are, in fact, sentential heads of gapped sentences that dropped their subjects, objects, or oblique objects to modify these dropped constituents. These derivations produce two types of participles according to the grammatical function of the dropped constituent: *subject* and *object participles* (see Underhill [15]).

  Derivations with *-yAn* and *-yAsH* may only produce subject participles, as illustrated in (24):

(24) a. Köşede duran adamı tanıyor musun?
`corner+LOC stand+PART man+ACC know+PROG QUES+2SG`
‘Do you know the man standing at the corner?’

b. övülesi adam
`praise+PART man`
'man deserving praise'

c. elleri öpülesi kadın
`hand+3PL+3SG kiss+PART woman`
'woman whose hands worth kissing'

Derivations using -*yAcAk* may produce both types of participles, whereas the ones with -*dHk* may only produce object participles. Consider example sentences in (25):

(25) a. Paketi alacak çocuk henüz gelmedi.
`packet+ACC take+PART boy yet come+NEG+PAST+3SG`
'The boy who will take the packet has not come yet.'

b. Gökhan'ın okuduğu kitabı ben daha önce
`Gökhan+GEN read+PART+3SG book+ACC I before`
okumuştum.
`read+NARR+PAST+1SG`
'I read the book that Gökhan is reading before.'

On the contrast, the qualitative adjectives derived form verbal with -*yHcH* are not heads of gapped sentences, e.g., *yazıcı* (*printer*). Note that as used in *tanıdık kişi* (*known person*), *bildik biri* (*known person*), and *giyecek elbise* (*dress to wear*) not all participles derived using -*dHk* and *yAcAk* are heads of gapped sentences.[8] These are the idiomatic usages of participles.

Derivation without using a suffix is also possible, e.g.,

(27) – bilir 'that cannot come',
– okur yazar 'that reads and writes',
– donmuş 'that is frozen'.

[8] Although the form *predicative verb+dHk* is not productive (i.e., only some of the verbs may conform to it), its negated form is generally applicable to all predicative verbs, as used in the following:

(26) a.
O kitap için sormadık dükkan bırakmadık.
That book for ask+NEG+PART shop leave+NEG+PAST+1PL
'We didn't left any shop that we didn't ask that book.'

b.
Çalmadık kapı kalmadı.
knock+NEG+PART door exist+NEG+PAST+3SG
'We consulted everyone.'

Only object participles derived using *-dHk* and *-yAcAk* take possessive suffix, since the subject may be missing in the subordinate clause (see the following example).

Consider the feature structure for *bilmediğim* (*that I don't know*), as used in (28):[9]

(28) Bilmediğim yemekleri hiçbir zaman yemem.
`know+NEG+PART+P1SG dish+3PL+ACC never eat+NEG+ARST+1SG`
'I never eat dishes that I don't know.'

$$
{}_{\textit{derived qualitative adj}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{adjective}\\ \text{SUB} & \text{qualitative}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \boxed{1}\\ \text{FORM} & \text{derived}\\ \text{DERV-SUFFIX} & \text{“dık”}\\ \text{POSS} & \text{1sg}\end{array}\right]\\
\text{SYN} & \left[\begin{array}{ll}\text{SUBCAT} & \boxed{2}\\ \text{MODIFIES} & \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{array}\right]\end{array}\right]\end{array}\right]\\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & f_{dık}(\boxed{4})\\ \text{ROLES} & \boxed{3}\end{array}\right]\\
\text{PHON} & \text{“bilmediğim”}
\end{array}\right]
$$

[9] The constraint structures of subcategorization information for the verb *bil-* are given on page 32.

$$
\underset{\textit{lexical predicative verb}}{\boxed{1}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predivative}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“bil”}\\ \text{FORM} & \text{lexical}\\ \text{SENSE} & \text{neg}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\left\langle \begin{matrix}\boxed{5}\begin{bmatrix}\text{SYN-ROLE} & \text{subject}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{constraint_1\right\}\end{bmatrix},\\ \boxed{6}\begin{bmatrix}\text{SYN-ROLE} & \text{dir-obj}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\begin{matrix}constraint_2,\\ constraint_3,\\ constraint_4,\\ constraint_5\end{matrix}\right\}\end{bmatrix}\end{matrix}\right\rangle\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{4}\ \text{\#bil-(to know something)}\\ \text{ROLES} & \boxed{3}\begin{bmatrix}\text{AGENT} & \boxed{5}\\ \text{THEME} & \boxed{6}\end{bmatrix}\end{bmatrix}\\
\text{PHON} & \text{“bil”}
\end{bmatrix}
$$

3.6 Adverbials

This section describes the representation of adverbials in our lexicon. These are words that modify or add to the meaning of verbs (and verbal forms), adjectives, and adverbials in various ways, e.g., direction, manner, temporality, etc. (see Ediskun [3]). As depicted in Figure 3.17, adverbials are divided into five subcategories, whose details are given in Figure 3.18.

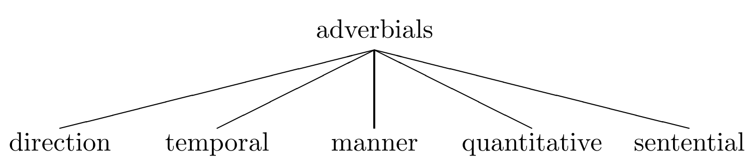

Figure 3.17: Subcategories of adverbials.

Each adverb has the following additional feature, which describes whether the adverb in consideration is in questional form or not. For instance, adverbs *neden* (*why*) and *nasıl* (*how*) are in questional form.

maj	*min*	*sub*	*ssub*
adverbial	direction		
	temporal	point-of-time	
		time-period	fuzzy
			day-time
			season
	manner	qualitative	
		repetition	
	quantitative	approximation	
		comparative	
		superlative	
		excessiveness	
	sentential		

Figure 3.18: Lexicon categories of adverbials.

$$_{adverbial}\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{QUESTIONAL} & +/- \text{ (default: } -)\end{bmatrix}\end{bmatrix}$$

3.6.1 Direction Adverbs

As the name implies, direction adverbs modify verbs and verbal forms by specifying direction. The following are examples of direction adverbs: *dışarı* (*out*), *beri* (*here*), *içeri* (*in*), *geri* (*back*), *karşı* (*opposite*).

Consider the feature structure of the direction adverb *dışarı* (*out*), as used in (29):

(29) Dışarı mı çıkıyorsun?
`out QUES get+PROG+1SG`
'Are you getting out?'

$$_{direction\ adv}\begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{adverbial}\\ \text{MIN} & \text{direction}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“dışarı”}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#dışarı-(out)}\end{bmatrix}\\ \text{PHON} & \text{“dışarı”}\end{bmatrix}$$

3.6.2 Temporal Adverbs

Temporal adverbs specify the point of time and limit the period of states, actions, and processes. As shown in Figure 3.19 temporal adverbs comprise *point-of-time* and *time-period adverbs*.

temporal adverbs

point-of-time   time-period

Figure 3.19: Subcategories of temporal adverbs.

Point-of-Time Adverbs

There are two forms of point-of-time adverbs: *lexical* and *derived*. The following two sections describe these with examples.

Lexical The following are point-of-time adverbs in lexical form: *dün* (*yesterday*), *bugün* (*today*), *şimdi* (*now*), *demin* (*a moment ago*), *önce* (*before*), *önceden* (*beforehand*).

Derived This form of adverbs are derived from verbs using suffixes -*yHp* and -*yHncA*. The derivation with -*yHp* produces adverbs that state a subordinate action that happens simultaneously or in sequence with the main action in the sentence. The other type of adverbs state an action that happens in sequence with the main action. Consider the following examples:

(30) a. Bu soruyu, konuyu anlayıp çözmek lazım.
`this question+ACC topic+ACC understand+ADV solve+INF needed+PRES+3SG`
'It is first needed to understand the topic and then to solve this question.'

b. Bu akşam kitap okuyup dinlenecektim.[10]
`this evening book read+ADV rest+FUT+PAST+1SG`
'This evening I was going to read a book and rest.'

In the first sentence, the adverb, *anlayıp*, states a subordinate action that is performed before the main action. In the latter one, however, the two actions happen simultaneously.

Each derived point-of-time adverb has the following additional features, which give the derivation suffix, subcategorization information and thematic roles of the verb involved in the derivation.

[10] This example is due to Underhill [15].

$$
{}_{\textit{derived point-of-time adv}}\left[\begin{array}{ll}
\text{MORPH} & \left[\text{DERV-SUFFIX} \quad \text{“yınca”/“yıp”}\right] \\
\text{SYN} & \left[\text{SUBCAT} \quad \textit{subcat}\right] \\
\text{SEM} & \left[\text{ROLES} \quad \textit{roles}\right]
\end{array}\right]
$$

Consider the feature structure for *bitince* (*when it ends*), as used in (31):

(31) a. Toplantı bitince, konuşmacıya bu konundaki
`meeting end+ADV speaker+DAT this subject+LOC+REL`
fikrimi açıkladım.
`opinion+P1SG+ACC explain+PAST+1SG`
‘When the meeting ended, I explained my opinion about this subject to the speaker.’

b. Odanı toplaman bitince hemen
`room+P2SG+ACC tidy up+INF+P2SG finish+ADV immediately`
yatmanı istiyorum.
`go to bed+INF+P2SG+ACC want+PROG+1SG`
‘I want you to go to bed as soon as you finish tidying up your room.’

$$
{}_{\textit{derived point-of-time adv}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{adverbial} \\ \text{MIN} & \text{temporal} \\ \text{SUB} & \text{point-of-time} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \boxed{1} \\ \text{FORM} & \text{derived} \\ \text{DERV-SUFFIX} & \text{“yınca”} \end{array}\right] \\
\text{SYN} & \left[\text{SUBCAT} \quad \boxed{2}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \text{f}_{\textit{yınca}}(\boxed{4}) \\ \text{ROLES} & \boxed{3} \end{array}\right] \\
\text{PHON} & \text{“bitince”}
\end{array}\right]
$$

$$\boxed{1}\ {}_{\textit{lexical predicative verb}}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predicative}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{STEM} & \text{``bit''}\\ \text{SENSE} & \text{pos}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\left\langle \boxed{4}\begin{bmatrix}\text{SYN-ROLE} & \text{subject}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\begin{matrix} constraint_1, \\ constraint_2\end{matrix}\right\}\end{bmatrix}\right\rangle\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{4}\ \text{\#bit-(to end)}\\ \text{ROLES} & \boxed{3}\begin{bmatrix}\text{AGENT} & \boxed{5}\end{bmatrix}\end{bmatrix}\\ \text{PHON} & \text{``bit''}\end{bmatrix}$$

$${}_{constraint_1}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \left\{\text{noun, pronoun}\right\}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}\end{bmatrix}$$

$${}_{constraint_2}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{sentential}\\ \text{SUB} & \text{act}\\ \text{SSUB} & \text{infinitive}\\ \text{SSSUB} & \text{ma}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{CASE} & \text{nom}\end{bmatrix}\end{bmatrix}$$

Time-Period Adverbs

As Figure 3.20 shows, time-period adverbs are subdivided into three categories: *fuzzy*, *day-time*, and *season adverbs.*

Fuzzy There are two forms of fuzzy time-period adverbs: *lexical* and *derived.* In the following two sections we will describe these forms with examples.

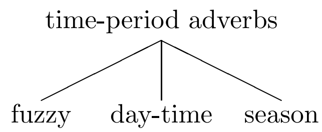

Figure 3.20: Subcategories of time-period adverbs.

Lexical The following are examples of this form of fuzzy time-period adverbs: *dakikalarca* (*for minutes*), *saatlerce/saatlerdir* (*for hours*).

Derived This form of adverbs are derived form verbs using the suffixes, *-yAlH* and *-ken*, as in

(32)	– sen geleli/gideli	'since the time you arrived/went',
	– biz gelirken	'while we are coming'.

Each derived fuzzy time-period adverb also has the following features. The derivation suffix is one of *-yAlH* and *-ken*. The other features give subcategorization information and semantic roles of the verb which are involved in the derivation process.

$$\left[\begin{array}{ll} \text{MORPH} & \left[\text{DERV-SUFFIX} \quad \text{“yalı”/“ken”}\right] \\ \text{SYN} & \left[\text{SUBCAT} \quad \textit{subcat}\right] \\ \text{SEM} & \left[\text{ROLES} \quad \textit{roles}\right] \end{array}\right]_{\textit{derived fuzzy time-period adv}}$$

Day-time *Sabahleyin* (*in the morning*), *sabahları* (*in the mornings*), *akşamları* (*in the evenings*), *gündüz* (*in the daytime*) and *gündüzleyin* (*in the daytime*) are examples of day-time time-period adverbs.

Season *Kışın* (*in the winter*) and *yazın* (*in the summer*) are two examples of season time-period adverbs.

3.6.3 Manner Adverbs

Manner adverbs describe the way and how actions, processes, and states develop. As depicted in Figure 3.21 manner adverbs are divided into two subcategories as *qualitative* and *repetition adverbs*, which are described next in detail.

manner adverbs

qualitative repetition

Figure 3.21: Subcategories of manner adverbs.

Qualitative Manner Adverbs

There are two forms of qualitative manner adverbs: *lexical* and *derived*. In the next sections, we will describe these forms in detail with examples.

Lexical The following are examples of qualitative manner adverbs in lexical form: *birden* (*suddenly*), *çabuk* (*fast*), *çabucak* (*fast*), *şöyle* (*like that*), *nasıl* (*how*).

Derived Each derived qualitative manner adverb has the following additional features, in which derivation suffix, subcategorization information and semantic roles are present. Derivation suffix feature may take one of the following values: *-cAsHnA*, *-mAksHzHn*, *-mAdAn*, *-yAmAdAn*, *-yArAk*, and *-cA*.

$$
{}_{\textit{derived qualitative adv}}\begin{bmatrix} \text{MORPH} & \begin{bmatrix} \text{DERV-SUFFIX} & \textit{derv-suffix} \end{bmatrix} \\ \text{SYN} & \begin{bmatrix} \text{SUBCAT} & \textit{subcat} \text{ (default: none)} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{ROLES} & \textit{roles} \text{ (default: none)} \end{bmatrix} \end{bmatrix}
$$

There are two types of derivations to this form of adverbs:

- Adjectival derivation: This derivation uses the suffix *-cA*, as in *akıllıca* (*intelligently*), *hızlıca* (*fast*), and *aptalca* (*stupidly*). Consider the feature structure for the qualitative adverb *akıllıca* as used in (33):[11]

  (33) Bugün, oldukça akıllıca davrandın.
  `today rather intelligently behave+PAST+2SG`
  'You behaved rather intelligently today.'

[11] SYN | SUBCAT feature is co-indexed with that of *akıllı*, which is shown in the section on qualitative adjectives on page 42.

$$
{}_{\textit{derived qualitative adv}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{adverbial} \\ \text{MIN} & \text{manner} \\ \text{SUB} & \text{qualitative} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \boxed{\text{“akıllı”}} \\ \text{FORM} & \text{derived} \\ \text{DERV-SUFFIX} & \text{“ca”} \end{array}\right] \\
\text{SYN} & \left[\text{SUBCAT} \quad \text{none}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \mathrm{f}_{ca}(\mathrm{f}_{lı}(\#\text{akıl-(intelligence)})) \\ \text{ROLES} & \text{none} \end{array}\right] \\
\text{PHON} & \text{“akıllıca”}
\end{array}\right]
$$

- Verb derivation: This derivation uses the suffixes *-cAsHnA*, *-mAksHzHn*, *-mAdAn*, *-yAmAdAn*, and *-yArAk*, as in the examples below:

(34)	– koşarcasına	‘as if running’,
	– görmeksizin	‘without seeing’,
	– gelmeden	‘without coming’,
	– göremeden	‘without seeing’,
	– gelerek	‘by coming’.

Repetition Manner Adverbs

As the name implies, this category of manner adverbs add repetition to the semantics of the verb and verbal forms. There are two forms of repetition manner adverbs, which are *lexical* and *derived*

Lexical *Tekrar* (*again*), *gene* (*again*), *sık* (*frequently*) are some examples of this form.

Derived The derivation to this form is only from verbs and uses the suffix *-dHkçA* as in:

(35)	– sen geldikçe	‘as you come’,
	– onlar konuştukça	‘as they talk’.

Each derived repetition adverb has the following additional feature structure, which has the derivation suffix, subcategorization information and thematic roles.

$$\textit{derived repetition adv}\begin{bmatrix} \text{MORPH} & \begin{bmatrix}\text{DERV-SUFFIX} & \text{“dıkça”}\end{bmatrix} \\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \textit{subcat}\end{bmatrix} \\ \text{SEM} & \begin{bmatrix}\text{ROLES} & \textit{roles}\end{bmatrix} \end{bmatrix}$$

3.6.4 Quantitative Adverbs

Quantitative adverbs modify the semantics of adjectivals, adverbials, and verbs in quantity. As shown in Figure 3.22, quantitative adverbs consist of four subcategories, for which many examples are given in the next sections.

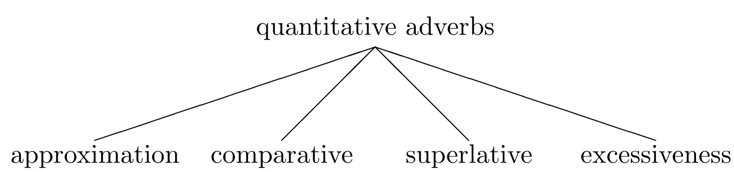

Figure 3.22: Subcategories of quantitative adverbs.

Approximation

Aşağı yukarı (*approximately*) and *hemen hemen* (*approximately*) are two examples of adverbs that are stating approximation.

Comparative

Daha (*more*) is the only member of this category.

Superlative

En (*most*) is the unique example of this category.

Excessiveness

The following are some examples of quantitative adverbs stating excessiveness: *çok* (*very*), *pek/gayet* (*very*), *fazla* (*too much*), *az/biraz* (*little*).

3.6.5 Sentential Adverbs

Sentential adverbs can only modify verbs and verbal forms. The following are some examples of sentential adverbs: *evet* (*yes*), *yok* (*no*), *öyle* (*so*), *elbette* (*certainly*), *gerçekten* (*really*), *daima* (*always*), *neden* (*why*).

3.7 Verbs

This section describes the representation of verbs in our lexicon with an emphasis on argument structures and thematic roles. Verb is the head of sentence, hence it is the most important constituent. It describes a state, action, or process [16]. As shown in Figure 3.23, verbs are divided into three categories as *predicative*, *existential*, and *attributive verbs*.

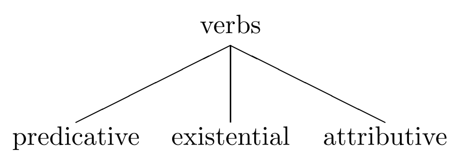

Figure 3.23: Subcategories of verbs.

Each verb in the lexicon has the following additional features, which represent morhosyntactic, syntactic, and semantic information. *none* is the default value for all of the features.

$$
{}_{verb}\left[\begin{array}{ll}
\text{MORPH} & \left[\begin{array}{ll}\text{TAM2} & tam2 \\ \text{COPULA} & 1/2 \\ \text{AGR} & agr\end{array}\right] \\
\text{SYN} & \left[\text{SUBCAT} \quad \left\langle role_1, \ldots, role_i, \ldots, role_n \right\rangle\right] \\
\text{SEM} & \left[\begin{array}{ll}\text{ROLES} & \left[\begin{array}{l}\text{AGENT} \\ \text{EXPERIENCER} \\ \text{PATIENT} \\ \text{THEME} \\ \text{RECIPIENT} \\ \text{CAUSER} \\ \text{ACCOMPANIER} \\ \text{SOURCE} \\ \text{GOAL} \\ \text{LOCATION} \\ \text{INSTRUMENT} \\ \text{BENEFICIARY} \\ \text{VALUE-DES}\end{array}\right]\end{array}\right]
\end{array}\right]
$$

There are four morphosyntactic features introduced (see Solak and Oflazer [13]). The MORPH | SENSE feature specifies whether the verb states a *positive* or *negative* predicate, attribute, etc. There are four possible tenses for attributive and existential verbs, which are also the possible second tenses for predicative verbs: *present*, *definite past*, *narrative past*, and *conditional forms*. This information is specified in MORPH | TAM2 feature. The feature MORPH | COPULA gives the usage of the suffix, *-dHr*, which states probability or definiteness. The last one represents the person suffix, whose possible values are *first*, *second*, and *third person singular*, and *plural persons*.

The subcategorization information, which we will describe later in detail, gives the valence of the verb for the *active* voice.[12]

---

[12] There are cases, in which the passive or causative voice of the verb gives a different sense than the active voice. In those cases, representation is configured accordingly, e.g.,

(36) a.

Kemal'i kapıya kadar geçirdik.
`Kemal+ACC door+DAT up to see off+PAST+1PL`
'We see Kemal off at the door.'

b.

İbrahim Ayşe'ye vuruldu.
`Ibrahim Ayşe+DAT fall in love+PAST+3SG`
'İbrahim fell in love with Ayşe.'

The feature SEM | ROLES describes the thematic roles of the arguments of the verb. These role fillers are the following (see Yılmaz [16]):

- agent,
- experiencer,
- theme,
- patient,
- causer,
- accompanier,
- recipient,
- goal,
- source,
- instrument,
- value designator,
- beneficiary,
- location.

The subcategorization information is given as a list of elements, each one describing an argument of the verb in question. Each such description consists of three features:

$$role_i \begin{bmatrix} \text{SYN-ROLE} & \textit{syn-role} \\ \text{OCCURRENCE} & \text{obligatory/optional} \\ \text{CONSTRAINTS} & \left\{ constraint_1, \ldots, constraint_j, \ldots, constraint_m \right\} \end{bmatrix}$$

The feature SYN-ROLE gives the argument type, which is one of the following:

- subject,
- direct object,

- agentive object,
- oblique objects (dative, ablative, locative)
- instrumental object,
- beneficiary object,
- value designator.

The second feature describes whether the occurrence of the argument is *obligatory* or *optional*. The last feature gives a list of constraints on the argument in consideration.

Elements in the subcategorization list are co-indexed with corresponding thematic role fillers according to the verb in consideration, i.e., there is a mapping from grammatical functions to thematic roles. For example, direct object is generally co-indexed with patient or theme.

The types of constraint structures are different for subject and (direct, oblique, and agentive) objects, instrumental object, value designator, and beneficiary object. Each structure will be described in turn:

- Constraint structures for subject, direct, oblique and agentive objects: The type of constraint structures for subject, direct, oblique, and agentive objects is given below. This feature structure gives constraints on the category, which is *nominal* in the most general case, a number of morphosyntactic and semantic properties of the argument.

$$
\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{nominal} \\ \text{MIN} & \textit{min} \\ \text{SUB} & \textit{sub} \\ \text{SSUB} & \textit{ssub} \\ \text{SSSUB} & \textit{sssub} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \textit{stem} \\ \text{CASE} & \textit{case} \\ \text{POSS} & \textit{poss} \\ \text{AGR} & \textit{agr} \end{array}\right] \\
\text{SEM} & [\,]
\end{array}\right]_{constraint_j}
$$

The subject never takes a case marking, i.e., it is in *nominative* case. There are cases that morphosyntactic features, other than the case, should be constrained, as well, as illustrated below:

(37) a. İstanbul'u sel aldı.
`Istanbul+ACC be flooded+PAST+3SG`
'İstanbul is flooded.'

b. Çocuk kafayı yedi.
`boy get mentally deranged+PAST+3SG`
'The boy got mentally deranged.'

In (37a), in addition to the case, the stem and the possessive marker are required to be *sel* and *none*, respectively. In the second sentence, however, the requirements are the following: the stem of the direct object is *kafa*; it has *accusative* case and *3sg* agreement markers, and it is not possessive-marked.

Semantic constraints can also be posed in these structures. For example, the verb sense *kafayı ye* (*to get metally deranged*) requires the subject to be human.

The direct object may be in *nominative* or *accusative* cases, while oblique objects are in *dative*, *ablative*, and *locative* cases.

The agentive object is in *ablative* case, and its stem is *taraf* with a suitable possessive marker. An example sentence is given in (38):

(38) Sorun bizim tarafımızdan çözüldü.
`problem us+GEN by solve+PASS+PAST+3SG`
'The problem is solved by us.'

- Constraint structrures for instrumental object: The following are the constraint structures for the instrumental object. There are two possible types for this argument. The first type is for nominals, which are *instrumental* case-marked. The second is for post-positional phrases, whose heads are the post-position *ile*:[13]

$$
constraint_j \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \textit{min} \\ \text{SUB} & \textit{sub} \\ \text{SSUB} & \textit{ssub} \\ \text{SSSUB} & \textit{sssub} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{ins} \end{bmatrix} \\ \text{SEM} & [] \end{bmatrix}
$$

[13] There are two additional forms with the nominals *saye*+POSS+LOC and *aracılık*+POSS+INS (*aracılık*+POSS *ile*). These can be represented with the structures introduced above by imposing proper morphosyntactic constraints, e.g., MORPH | STEM = "saye", MORPH | CASE = loc, MORPH | AGR = 3sg. But we will omit these forms.

$$\mathit{constraint}_j \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{post-position} \\ \text{MIN} & \text{ins-subcat} \end{array}\right] \\ \text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \text{``ile''} \end{array}\right] \\ \text{SEM} & [\,] \end{array}\right]$$

- Constraint structures for value designator: There are two forms in a sentence to describe a value designator. The first form uses a nominal, which is *dative* case-marked. The second uses a post-positional phrase whose head is *için*, as used in (39):[14]

  (39) Oralarda 10 dolar için adam öldürüler.
  `there+LOC 10 dolar for man kill+ARST+3PL`
  They will kill you for 10 dollars there.

  Thus, the two feature structures that are introduced for instrumental object can be used for the value designator by replacing the values of case, stem, and the minor category features with *dative*, *için*, and *nom-subcat* respectively.

- Constraint structures for beneficiary object: The feature structure below is for the beneficiary object, which is a post-positional phrase whose head is the post-position, *için*:

$$\mathit{constraint}_j \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{post-position} \\ \text{MIN} & \text{nom-subcat} \end{array}\right] \\ \text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \text{``için''} \end{array}\right] \\ \text{SEM} & [\,] \end{array}\right]$$

  Furthermore, the oblique object case-marked as *dative* can be mapped to the beneficiary, as depicted in the following example:

  (40) Annesi, çocuğa uyumadan önce kitap okudu.
  `mother+P1SG boy+DAT sleep+INF+ABL before book read+PAST+3SG`
  'His mother read book for the boy before he slept.'

As mentioned above, the subcategorization information for verbs in lexical form is given as a list, in which each element gives constraints on an argument of the verb in consideration. Since the members of other categories in lexical form, such as common nouns, qualitative adjectives, and post-positions, cannot have more than one argument, just the constraint lists for one complement are given.

In the following sections we will describe the subcategories of verbs in detail.

[14] This example is due to Yılmaz [16].

3.7.1 Predicative Verbs

Predicative verb category comprises the verbs that are not existential or attributive. There are two forms of predicative verbs, which are *lexical* and *derived*. These forms are described in the next sections.

Each predicative verb has the following additional morphosyntactic features:

$$
{}_{\textit{predicative verb}}\left[\begin{array}{ll}
\mathrm{MORPH} & \left[\begin{array}{lll}
\mathrm{SENSE} & \mathrm{pos/neg} & \\
\mathrm{TAM1} & \textit{tam1} & \text{(default: none)} \\
\mathrm{COMP} & \textit{comp} & \text{(default: none)} \\
\mathrm{PASSIVE} & +/- & \text{(default: } -\text{)} \\
\mathrm{RECIPROCAL} & +/- & \text{(default: } -\text{)} \\
\mathrm{REFLEXIVE} & +/- & \text{(default: } -\text{)} \\
\mathrm{CAUSATIVE} & n & \text{(default: 0)}
\end{array}\right]
\end{array}\right]
$$

The first tense-aspect-mood marker is specified in MORPH | TAM1 feature, for which there are ten possible values: *present*, *definite past*, *narrative past*, *future*, *aorist*, *progressive*, *conditional*, *optative*, *necessitative*, and *imperative*. If the verb is a compound one, the compounding suffix is given in MORPH | COMP feature, whose value is one of *-yAbil*, *-yHver*, *-yAdur*, *-yAkoy*, *-yAkal*, and *-yAyaz*. The last four features represent the voice of the verb. The value n represents a positive integer number, which denotes the level of causation (see Solak and Oflazer [13]).

Lexical

This form of predicative verbs are present in the lexicon as lexical entries mainly consisting of subcategorization information and thematic roles. The following are example predicative verbs in lexical form:

(41)
- ye- ‘eat’,
- iç- ‘drink’,
- gör- ‘see’,
- hediye et- ‘give present’,
- kafayı ye- ‘get mentally deranged’,
- rüşvet ye- ‘receive bribe’.

Some of the predicative verbs consist of more than one word, e.g., *kafayı ye-* (*get mentally deranged*), *rezil et-* (*disgrace*), *rezil ol-* (*be disgraced*), *kavga et-* (*quarrel*), some of which are constructed with the auxiliary verbs *et-* and *ol-*. The verbs whose first constituents are not nominals are taken as separate compound verbs, whereas there are two cases for the ones whose first constituents are nominals. In the first case, such constituents are not subject to inflections as in (42a):

(42) a. *Biz yine de hediyemizi ederiz.

*Biz	yine de	hediyemizi	ederiz.
`we`	`anyway`	`present+1PL+ACC`	`do+ARST+1PL`

b. Biz gerekirse kavgamızı ederiz.

Biz	gerekirse	kavgamızı	ederiz.
`we`	`if needed`	`fight+1PL+ACC`	`do+ARST+1PL`

'If needed, we will fight.'

This type of verbs are taken separately as compound verbs. In the latter case, as in (42b), such constituents are subject to inflection, which are taken as a different sense of the main verb, and the first constituent is given as an object in the argument structure. For example, *kavga et-* (*quarrel*) is represented as a sense of *et-*, and *kavga* (*quarrel*) is the direct object of this sense.

We will give feature structures for four senses of the verb, *ye-*, which are the following:

1. eat something,
2. eat from something,
3. get mentally deranged,
4. be unfair.

The following is the feature structure for the first sense, *eat something*, as used in (43):

(43)

Adam	çatalla	pastayı	yedi.
`man`	`fork+INS`	`pastry+ACC`	`eat+PAST+3SG`

'The man ate the pastry with fork.'

$$
\left[\begin{array}{lll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predicative}\end{array}\right] & \\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“ye”}\\ \text{FORM} & \text{lexical}\\ \text{SENSE} & \text{pos}\\ \text{TAM1} & \text{past}\\ \text{AGR} & \text{3sg}\end{array}\right] & \\
\text{SYN} & \left[\begin{array}{ll}\text{SUBCAT} & \left\langle\begin{array}{l}
\boxed{1}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{subject}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\mathit{constaint}_1\right\}\end{array}\right],\\
\boxed{2}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{dir-obj}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\begin{array}{l}\mathit{constaint}_2,\\ \mathit{constaint}_3\end{array}\right\}\end{array}\right],\\
\boxed{3}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{inst-obj}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\begin{array}{l}\mathit{constaint}_4,\\ \mathit{constaint}_5\end{array}\right\}\end{array}\right]
\end{array}\right\rangle\end{array}\right] & \\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & \text{\#ye-(to eat something)}\\ \text{ROLES} & \left[\begin{array}{ll}\text{AGENT} & \boxed{1}\\ \text{THEME} & \boxed{2}\\ \text{INSTRUMENT} & \boxed{3}\end{array}\right]\end{array}\right] & \\
\text{PHON} & \text{“yedi”} &
\end{array}\right]_{\mathit{lexical\ predicative\ verb}}
$$

$$
\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \left\{\text{noun, pronoun}\right\}\end{array}\right]\\
\text{MORPH} & \left[\text{CASE}\quad\text{nom}\right]\\
\text{SEM} & \left[\text{ANIMATE}\ +\right]
\end{array}\right]_{\mathit{constraint}_1}
$$

$$\mathit{constraint}_2 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \left\{\text{acc, nom}\right\} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{EDIBLE} & + \end{bmatrix} \end{bmatrix}$$

$$\mathit{constraint}_3 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{pronoun} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{acc} \end{bmatrix} \end{bmatrix}$$

$$\mathit{constraint}_4 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \left\{\text{noun, pronoun}\right\} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{ins} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INSTRUMENT} & + \end{bmatrix} \end{bmatrix}$$

$$\mathit{constraint}_5 \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{post-position} \\ \text{MIN} & \text{ins-subcat} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“ile”} \end{bmatrix} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{INSTRUMENT} & + \end{bmatrix} \end{bmatrix}$$

The following is the feature structure for the second sense, *eat from something*, as used in (44):[15]

(44) Adam çatalla pastadan yedi.
`man fork+INS pastry+ABL eat+PAST+3SG`
‘The man ate from the pastry with fork.’

The difference between the first and the second senses is that the patient, *pasta* (*pastry*), is the direct object in the former one, whereas, it is the oblique object in ablative case in the latter. Note that the second sense does not subcategorize for a direct object.

[15] The feature structure for subject and instrumental object are the same with those of previous example.

$$
_{\textit{lexical predicative verb}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{verb} \\ \text{MIN} & \text{predicative} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“ye”} \\ \text{FORM} & \text{lexical} \\ \text{SENSE} & \text{pos} \\ \text{TAM1} & \text{past} \\ \text{AGR} & \text{3sg} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \text{SUBCAT} & \left\langle \begin{array}{l}
\boxed{1} \begin{bmatrix} \text{SYN-ROLE} & \text{subject} \\ \text{OCCURRENCE} & \text{optional} \\ \text{CONSTRAINTS} & \left\{ \textit{constaint}_1 \right\} \end{bmatrix}, \\
\boxed{2} \begin{bmatrix} \text{SYN-ROLE} & \text{obl-abl} \\ \text{OCCURRENCE} & \text{optional} \\ \text{CONSTRAINTS} & \left\{ \textit{constaint}_2 \right\} \end{bmatrix}, \\
\boxed{3} \begin{bmatrix} \text{SYN-ROLE} & \text{inst-obj} \\ \text{OCCURRENCE} & \text{optional} \\ \text{CONSTRAINTS} & \left\{ \begin{array}{l} \textit{constaint}_3, \\ \textit{constaint}_4 \end{array} \right\} \end{bmatrix}
\end{array} \right\rangle \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#ye-(to eat from something)} \\ \text{ROLES} & \begin{bmatrix} \text{AGENT} & \boxed{1} \\ \text{THEME} & \boxed{2} \\ \text{INSTRUMENT} & \boxed{3} \end{bmatrix} \end{bmatrix} \\
\text{PHON} & \text{“yedi”}
\end{bmatrix}
$$

$$
_{\textit{constraint}_2}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \left\{ \text{noun, pronoun} \right\} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{CASE} & \text{abl} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{EDIBLE} & + \end{bmatrix}
\end{bmatrix}
$$

The following is the feature structure for the third sense of *ye-*, *get mentally deranged*, as shown in (45):

(45) Cüneyt, okulda çok çalışmaktan
`Cüneyt school+LOC too much working+ABL`
kafayı yedi.
`get mentally deranged+PAST+3SG`
'Cüneyt got mentally deranged from too much working at the school.'

Note that the direct object has to be *kafayı*, and it is not a semantic role filler.

$$
\textit{lexical predicative verb}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{verb} \\ \text{MIN} & \text{predicative} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“ye”} \\ \text{FORM} & \text{lexical} \\ \text{SENSE} & \text{pos} \\ \text{TAM1} & \text{past} \\ \text{AGR} & \text{3sg} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \text{SUBCAT} & \left\langle \begin{array}{l} \boxed{1}\begin{bmatrix} \text{SYN-ROLE} & \text{subject} \\ \text{OCCURRENCE} & \text{optional} \\ \text{CONSTRAINTS} & \left\{ constaint_1 \right\} \end{bmatrix}, \\ \begin{bmatrix} \text{SYN-ROLE} & \text{dir-obj} \\ \text{OCCURRENCE} & \text{obligatory} \\ \text{CONSTRAINTS} & \left\{ constaint_2 \right\} \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#ye-(to get mentally deranged)} \\ \text{ROLES} & \begin{bmatrix} \text{EXPERIENCER} & \boxed{1} \end{bmatrix} \end{bmatrix} \\
\text{PHON} & \text{“yedi”}
\end{bmatrix}
$$

$$
constraint_1\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \left\{ \text{noun, pronoun} \right\} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{HUMAN} & + \end{bmatrix}
\end{bmatrix}
$$

$$constraint_2 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“kafa”} \\ \text{CASE} & \text{acc} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \end{bmatrix} \end{bmatrix}$$

The feature structure for the fourth sense of *ye-* is given below, in which the direct object, *hak*, is optionally *accusative* case-marked, as below:

(46) a. Oğuz hep hak yiyor.
`Oğuz always be unfair+PROG+3SG`
‘Oğuz is always unfair.’

b. Oğuz başkalarının da haklarını yedi.
`Oğuz others+GEN too be unfair+PAST+3SG`
‘Oğuz was unfair to the others, too.’

$$
{}_{\textit{lexical predicative verb}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predicative}\end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“ye”}\\ \text{FORM} & \text{lexical}\\ \text{SENSE} & \text{pos}\\ \text{TAM1} & \text{past}\\ \text{AGR} & \text{3sg}\end{array}\right] \\
\text{SYN} & \left[\begin{array}{ll}\text{SUBCAT} & \left\langle \begin{array}{l} \boxed{1}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{subject}\\ \text{OCCURRENCE} & \text{optional}\\ \text{CONSTRAINTS} & \left\{\mathit{constaint}_1\right\}\end{array}\right], \\ \boxed{2}\left[\begin{array}{ll}\text{SYN-ROLE} & \text{dir-obj}\\ \text{OCCURRENCE} & \text{obligatory}\\ \text{CONSTRAINTS} & \left\{\mathit{constaint}_2\right\}\end{array}\right]\end{array}\right\rangle\end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & \text{\#ye-(to be unfair)}\\ \text{ROLES} & \left[\begin{array}{ll}\text{AGENT} & \boxed{1}\\ \text{THEME} & \boxed{2}\end{array}\right]\end{array}\right] \\
\text{PHON} & \text{“yedi”}
\end{array}\right]
$$

$$
{}_{\mathit{constraint}_1}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \left\{\text{noun, pronoun}\right\}\end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll}\text{CASE} & \text{nom}\end{array}\right]
\end{array}\right]
$$

$$
{}_{\mathit{constraint}_2}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“hak”}\\ \text{CASE} & \left\{\text{acc, nom}\right\}\end{array}\right]
\end{array}\right]
$$

Derived

This form of verbs are derived from nominals and adjectivals using the suffixes *-lAn* and *-lAş*. Each derived predicative verb has the following additional feature, which gives the derivation suffix.

$$\left[\text{MORPH} \quad \left[\text{DERV-SUFFIX} \quad \text{“lan”/“laş”}\right]\right]_{\textit{derived verbal}}$$

There are two types of derivations to predicative verbs:

- Nominal derivation: This derivation uses the suffixes *-lAn* and *-lAş*. The following are some examples of predicative verbs derived form nominals:

  (47) – taşlaş- ‘turn into stone’,
  – ağaçlandır- ‘plant trees in an area’,
  – sinirlen- ‘get nervous’.

  Consider the feature structure for *sinirlen-*, as used in (48):

  (48) Tembellik etmen beni çok sinirlendiriyor!
  `laziness do+INF+P2SG me+ACC very make angry+PROG+3SG`
  ‘Your laziness is making me very angry!’

$$\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{predicative}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \boxed{1}\\ \text{FORM} & \text{derived}\\ \text{DERV-SUFFIX} & \text{“lan”}\\ \text{SENSE} & \text{pos}\\ \text{TAM1} & \text{prog1}\\ \text{CAUSATIVE} & 1\end{array}\right]\\
\text{SYN} & \left[\text{SUBCAT} \quad \boxed{2}\ \text{none}\right]\\
\text{SEM} & \left[\begin{array}{ll}\text{CONCEPT} & \text{f}_{lan}(\boxed{3})\\ \text{ROLES} & \text{none}\end{array}\right]\\
\text{PHON} & \text{“sinirlendiriyor”}
\end{array}\right]_{\textit{derived predicative verb}}$$

$$\boxed{1}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{array}\right]\\
\text{MORPH} & \left[\begin{array}{ll}\text{STEM} & \text{“sinir”}\\ \text{FORM} & \text{lexical}\end{array}\right]\\
\text{SYN} & \left[\text{SUBCAT} \quad \boxed{2}\ \text{none}\right]\\
\text{SEM} & \left[\text{CONCEPT} \quad \boxed{3}\ \text{\#sinir-(anger)}\right]\\
\text{PHON} & \text{“sinir”}
\end{array}\right]_{\textit{lexical common}}$$

- Adjectival derivation: This derivation uses the same suffixes. The following are some examples of predicative verbs derived from adjectivals: *iyileş-* (*recover from illness*), *uzaklaş-*, (*go away from*), *yaralan-* (*be hurted*).

3.7.2 Existential Verbs

This category of verbs consists of only *var* (*existent*) and *yok* (*nonexistent*), which state existence and non-existence in sentences, respectively. Two example sentences are given in (49):

(49) a. Masamda kağıt ve kalem var.
`table+P1SG+LOC paper and pencil existent+PRES+3SG`
'There are paper and pencil on my table.'

b. Bugün yapacak fazla işim yok.
`today do+PART much work+P1SG nonexistent+PRES+3SG`
'I don't have much work to do today.'

3.7.3 Attributive Verbs

Attributive verbs state properties of entities. This category consists of verbs in *lexical* and *derived* forms, which are described in the next sections.

Lexical

The only attributive verb that is in lexical form is *değil* (*not*). This verb makes the sentences negative whose heads, otherwise, are existential or derived attributive verbs, as shown in (50):

(50) a. Onun bisikleti kırmızıydı.
`his bicycle+P3SG red+PAST+3SG`
'His bicycle was red.'

b. Onun bisikleti kırmızı değildi.
`his bicycle+P3SG red NOT+PAST+3SG`
'His bicycle was not red.'

Derived

There are three ways to derive attributive verbs: from nominals, adjectivals, and post-positions. Attributive verbs in derived form have the following additional feature giving the derivation suffix, whose value is *none*, since none of the three derivations uses a suffix:

$$\left[\text{MORPH} \quad \left[\text{DERV-SUFFIX} \quad \text{none}\right]\right]_{\textit{derived attributive verb}}$$

There are three types of derivations to attributive verbs:

- Nominal derivation: The sentences below use this type of verb forms:

  (51) a. O yediğin benim elmamdı.
  `that eat+PART+P2SG my apple+P1SG+PAST+3SG`
  'It was my apple that you ate.'

  b. Bu sütün son kullanma tarihi dünmüş.
  `this milk+GEN last usage+P3SG date yesterday+NARR+3SG`
  'The expiry date of this milk was yesterday.'

- Adjectival derivation: The sentences below give some examples of attributive verbs derived from adjectivals:

  (52) a. Hızlı yazmakta oldukça becerikliyim.
  `fast write+INF+LOC very skillful+PRES+1SG`
  'I am very skillful in writing fast.'

  b. Sen kaçıncısın?
  `you in what rank+PRES+2SG`
  'What is your rank?'

  Consider the following feature structure for *borçluyum*, as used in (53), which is derived from the qualitative adjective *borçlu* (*that owing debt*). Note that *borçlu* is also derived from the common noun, *borç* (*debt*):[16]

  (53) Başarımı çok çalışmama borçluyum.
  `success+P1SG+ACC very much work+INF+DAT debtor+PRES+1SG`
  'It was my hard working that brought my success.'

[16] This example derivation considers only one sense of *borç*. This process is repeated for all of the senses of this noun regardless of the semantics of the derivation with the suffixes used. Furthermore, if the morphological processor allows a derivation starting from the adjective *borçlu*, this path is followed, as well.

$$ {}_{\textit{derived attributive verb}}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{verb}\\ \text{MIN} & \text{attributive}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \boxed{1}\\ \text{FORM} & \text{derived}\\ \text{AGR} & \text{1sg}\\ \text{TAM2} & \text{pres}\\ \text{DERV-SUFFIX} & \text{none}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{f}_{none}(\boxed{3})\end{bmatrix}\\ \text{PHON} & \text{“borçluyum”}\end{bmatrix} $$

$$ \boxed{1}\ {}_{\textit{derived qualitative adj}}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{adjectival}\\ \text{MIN} & \text{adjective}\\ \text{SUB} & \text{qualitative}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \boxed{4}\\ \text{FORM} & \text{derived}\\ \text{DERV-SUFFIX} & \text{“lı”}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\\ \text{MODIFIES} & \begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\end{bmatrix}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{3}\ \text{f}_{lı}(\boxed{5})\end{bmatrix}\\ \text{PHON} & \text{“borç+lı”}\end{bmatrix} $$

$$ \boxed{4}\ {}_{\textit{lexical common}}\begin{bmatrix} \text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“borç”}\\ \text{FORM} & \text{lexical}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \boxed{2}\left\{constraint_1, constraint_2, constraint_3\right\}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \boxed{5}\ \text{\#borç-(debt)}\end{bmatrix}\\ \text{PHON} & \text{“borç”}\end{bmatrix} $$

$$constraint_1 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \left\{\text{noun, pronoun}\right\} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{dat} \end{bmatrix} \end{bmatrix}$$

$$constraint_2 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{ma} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{dat} \end{bmatrix} \end{bmatrix}$$

$$constraint_3 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{yış} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{dat} \\ \text{POSS} & \neg\text{none} \end{bmatrix} \end{bmatrix}$$

- Post-position derivation: The following example demonstrates the derivation from post-position *sonra*:

(54) Sen benden sonrasın.
`you me+ABL after+PRES+2SG`
'You are after me.'

3.8 Conjunctions

This section describes the representation of conjunctions in our lexicon. Conjunctions are function words, i.e., they do not convey meaning when used alone. They are used to conjoin words, phrases, and sentences both syntactically and semantically (see Ediskun [3]). As shown in Figure 3.24, conjunctions are divided into three subcategories: *coordinating*, *bracketing* and *sentential conjunctions*.

The next three sections describe the subcategories of conjunctions with examples.

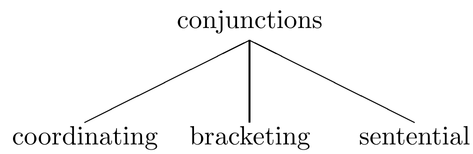

Figure 3.24: Subcategories of conjunctions.

3.8.1 Coordinating Conjunctions

The following are examples of coordinating conjunctions: *ile* (*and*), *ve* (*and*), *veya* (*or*), *ila* (*between . . . and*).

Consider the feature structure of the coordinating conjunction *ve* (*and*), as used in the example below:

(55) Bugün ve yarın hava bulutlu olacakmış.
`today and tomorrow weather cloudy be+FUT+NARR+3SG`
'They say, today and tomorrow the weather will be cloudy.'

$$
{}_{coordinating}\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{conjunction} \\ \text{MIN} & \text{coordinating} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“ve”} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#ve-(and)} \end{bmatrix} \\ \text{PHON} & \text{ve”} \end{bmatrix}
$$

3.8.2 Bracketing Conjunctions

Bracketing conjunctions are used in pairs. These have the following two semantic features. The first gives the polarity of the conjunction, e.g., the polarity of *ne . . . ne* (*neither . . . nor*) is negative, while it is positive for *hem . . . hem* (*both . . . and*). The second specifies how the two elements bracketed are connected.

$$
{}_{bracketing}\begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{POLARITY} & +/- \text{ (default: +)} \\ \text{CONNECTION} & \text{and/or (default: and)} \end{bmatrix} \end{bmatrix}
$$

The following are some examples of bracketing conjunctions: *gerek . . . gerek*(*se*) (*both . . . and*), *ne . . . ne* (*neither . . . nor*), *hem . . . hem* (*both . . . and*), *ya . . . ya* (*either . . . or*).

The following is the feature structure of the bracketing conjunction, *gerek . . . gerek* (*both . . . and*), as used in (56):

(56) Gerek Yücel gerek Uğur bugün çok hızlı koştular.
`both Yücel and Uğur today very fast run+PAST+3PL`
'Both Yücel and Uğur ran very fast today.'

$$
{}_{\textit{bracketing}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{conjunction}\\ \text{MIN} & \text{bracketing}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“gerek \ldots gerek”}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#gerek \ldots gerek-(both \ldots and)}\end{bmatrix}\\
\text{PHON} & \text{“gerek \ldots gerek”}
\end{bmatrix}
$$

3.8.3 Sentential Conjunctions

Sentential conjunctions conjoin sentences. *Ancak* (*but*), *çünkü* (*because*), *hatta* (*even*), *ama* (*but*), *nitekim* (*just as*), *eğer* (*if*), *yani* (*that is to say*), and *üstelik* (*furthermore*) are some examples of sentential conjunctions.

3.9 Post-positions

This section describes the representation of post-positions in our lexicon. Like conjunctions, post-positions are function words, i.e., they do not have meaning, unless they are used with nominals in order to construct post-positional phrases (see Ediskun [3]). As shown in Figure 3.25, post-positions are subdivided into six categories according to their subcategorization types (specifically, the case of the complement).

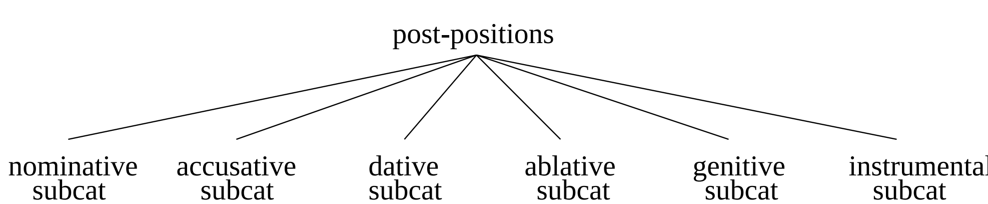

Figure 3.25: Subcategories of post-positions.

Each post-position also has the following feature, which gives the subcategorization information for only one argument, in contrast to the case in verbs, which accept a number of arguments, such as subject, direct object, etc. For this reason the subcategorization information of post-positions consists of just a list of constraints for only one argument.

$$\begin{bmatrix}\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \textit{subcat}\end{bmatrix}\end{bmatrix}_{\textit{post-position}}$$

In the next sections we will describe the subcategories and give examples for each of them.

3.9.1 Post-positions with Nominative Subcategorization

Post-positions belonging to this subcategory accept nominals in *nominative* case as complements. *Boyunca* (*along/during*), *takdirde* (*if*), *diye* (*named*), *için* (*for*) are examples of post-positions with nominative subcategorization.

The feature structure of the post-position, *için* (*for/because/in order to*), as used in (57), is given below, though the case of the complement is *genitive* for pronouns:

(57) a. Almayı unuttuğum kitaplar için odama
`take+INF+ACC forget+PART+P1SG book+3PL for room+P1SG+DAT`
tekrar gittim.
`again go+PAST+1SG`
'I went to my room again for the books that I forgot to take.'

b. Başarılı olabilmesi için çok çalışması
`succesfull be+ABIL+INF+P3SG for much work+INF+P3SG`
gerekiyor.
`needed+PROG+3SG`
'In order to be successful, he should work hard.'

$$\begin{bmatrix}\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{post-position}\\ \text{MIN} & \text{nom-subcat}\end{bmatrix}\\ \text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“için”}\end{bmatrix}\\ \text{SYN} & \begin{bmatrix}\text{SUBCAT} & \left\{\begin{matrix} constraint_1, constraint_2, constraint_3,\\ constraint_4, constraint_5, constraint_6\end{matrix}\right\}\end{bmatrix}\\ \text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#için-(for/because/in order to)}\end{bmatrix}\\ \text{PHON} & \text{“için”}\end{bmatrix}_{\textit{nom-subcat}}$$

$$constraint_1 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \end{bmatrix} \end{bmatrix}$$

$$constraint_2 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{pronoun} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{gen} \end{bmatrix} \end{bmatrix}$$

$$constraint_3 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{mak} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{POSS} & \text{none} \end{bmatrix} \end{bmatrix}$$

$$constraint_4 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{ma} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{POSS} & \neg\text{none} \end{bmatrix} \end{bmatrix}$$

$$constraint_5 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{yış} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \end{bmatrix} \end{bmatrix}$$

$$
constraint_6 \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{sentential} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{participle} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \text{CASE} & \text{nom} \\ \text{POSS} & \neg\text{none} \end{bmatrix} \end{bmatrix}
$$

3.9.2 Post-positions with Accusative Subcategorization

Post-positions belonging to this subcategory accept nominals in *accusative* case as complements. The following examples are post-positions belonging to this category: *aşkın* (*over*), *takiben* (*following*), *müteakiben* (*following*).

3.9.3 Post-positions with Dative Subcategorization

Post-positions belonging to this subcategory accept nominals in *dative* case as complements. The following examples are post-positions belonging to this category: *ait* (*belonging to*), *göre* (*according to*), *dek* (*until*), *karşın* (*in spite of*), *yönelik* (*aimed at*), *doğru* (*towards*), *ilişkin* (*related to*).

3.9.4 Post-positions with Ablative Subcategorization

Post-positions belonging to this subcategory accept nominals in *ablative* case as complements. *Dolayı* (*due to*), *ötürü* (*due to*), *itibaren* (*starting from*), *sonra* (*after*), and *önce* (*before*) are examples of post-positions with *ablative* subcategorization.

3.9.5 Post-positions with Genitive Subcategorization

Post-positions belonging to this subcategory accept nominals (specifically, pronouns) in *genitive* case as complements. *İle* (*with*) is an example of this type of post-positions.

3.9.6 Post-positions with Instrumental Subcategorization

Post-positions belonging to this subcategory accept nominals in *instrumental* case as complements. The following post-positions are examples of this category: *birlikte* (*together*), *beraber* (*together*).

Chapter 4

Operational Aspects of the Lexicon

Our lexicon provides necessary morphosyntactic, syntactic, and semantic information to NLP subsystems performing syntactic analysis, tagging, semantic disambiguation, etc.

The whole system consists of three main parts:

1. a morphological processor/analyzer,
2. a static lexicon, and
3. a module filtering the output according to the user's restrictions.

As depicted in Figure 4.1, the system receives a query form, which includes, at least, a surface form and other information acting as the restrictions on the output feature structures. The surface form is first directed to the morphological processor, which generates all possible interpretations (i.e., parses or lexical forms) and forwards these to the static lexicon. The static lexicon accesses feature structure database and retrieves syntactic and semantic information for the root words involved in the interpretations. Having unified the morphosyntactic information provided with corresponding syntactic and semantic information retrieved, the static lexicon outputs a list of feature structures. The final step in the process is the elimination of the feature structures which do not satisfy the user's restrictions.

In this way, the NLP subsystems using the lexicon do not need to interface with the morphological processor to obtain interpretations, rather they just provide the surface form and receive the corresponding feature structures containing morphosyntactic, syntactic, and semantic information.

In this chapter, we will first describe the interface to the lexicon. Section 4.2 describes how the system produces feature structures step by step by giving examples, and Section 4.3 mentions problems and limitations related with this task.

4.1 Interfacing with the Lexicon

We presented many examples of feature structures in Chapter 3 and will describe the method of producing those feature structures in the next section. In this section, we will mainly concentrate on how NLP subsystems can use our lexicon.

Our lexicon is a front end for a morphological analyzer. Given a surface form with restriction features, it generates all the morphosyntactic, syntactic, and semantic information for this surface form, that is it abstracts morphological analysis and associates syntactic and semantic information with each interpretation (see Figure 4.2).

The interface described above can be used by a syntactic analyzer for Turkish. Additionally, taggers and word sense disambiguators can employ our lexicon. Taggers need to set necessary constraints, which are generally on category and morphosyntactic features, in the query form. Consider the following example:

(58) a. evin kapısı
`house+GEN door+P3SG`
'door of the house'

b. senin evin
`you+GEN house+P2SG`
'your house'

In the two noun phrases above, the surface form *evin* exists with two different interpretations: in the first one, it is *genitive* case-marked and singular with no possessive marking, whereas in the second one it is *nominative* case-marked with *2sg* possessive marking. The ambiguity can be resolved with the help of morphological features, i.e., case or possessive markings.

Word sense disambiguation is also possible by making use of semantic features in the feature structures. For example, the two senses of the root word *kazma* (*stupid person* and *pickaxe*) can be resolved by setting the SEM | ANIMATE feature in the query form properly. Adding semantic features increases the accuracy of word sense disambiguation process. However, rather than adding arbitrary semantic features on demand, constructing an ontology describing concepts via a semantic network would be more useful.

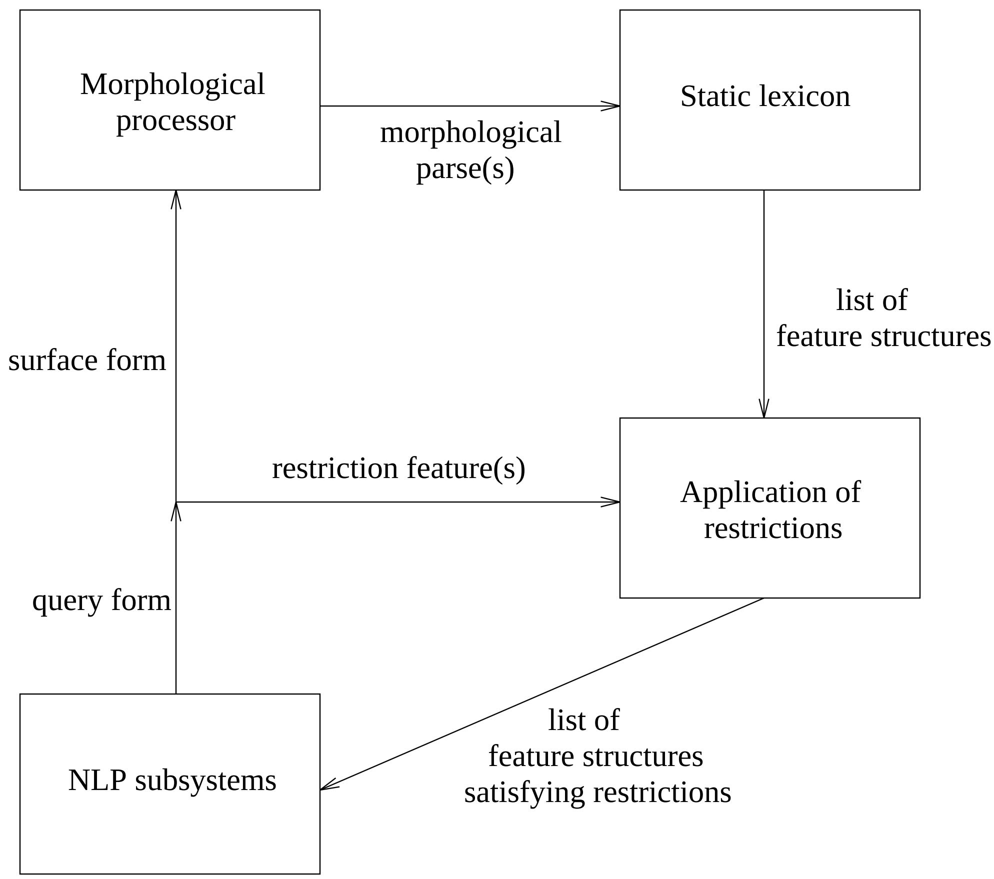

Figure 4.1: Data flow in the lexicon.

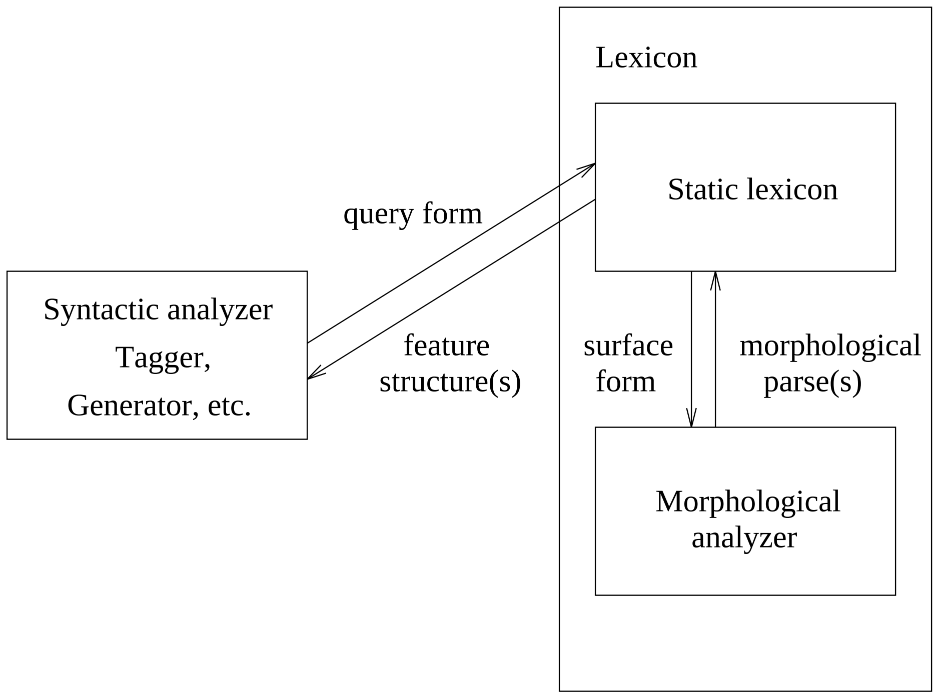

Figure 4.2: NLP subsystems interfacing with the lexicon.

Text generators for Turkish or transfer units to Turkish in machine translation systems can also make use of our lexicon to obtain information about root words. However, the SEM | CONCEPT feature may not be directly usable by transfer units, since the English definition in this feature is mostly human oriented.

The input query form is basically a feature structure, which contains two types of information: a surface form and a set of other features. The surface form guides the system in producing the feature structures, that is it is the actual input for the output of the lexicon. It is specified as the phonology information (the PHON feature) in the query form. The rest of the features are optional and act as restrictions on the output structures. In fact, the query form *subsumes* each of the actual output feature structures. Any set of features can be specified in the query form provided that they are *consistent* and *appropriate* for the intended structure.

The process of eliminating or filtering the output feature structures that do not satisfy the restrictions in the query form is the last step in the whole process.

Consider the following query form placing morphosyntactic and semantic restrictions on the surface form *ekimde*, that is the root word should not be possessive-marked, and its semantics should state temporality.

$$
{}_{\textit{query form}}\begin{bmatrix}
\text{MORPH} & \begin{bmatrix}\text{POSS} & \text{none}\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{TEMPORAL} & +\end{bmatrix} \\
\text{PHON} & \text{“ekimde”}
\end{bmatrix}
$$

According to the morphological processor, there are two interpretations of *ekimde*:

1. *Ekimde* (*in October*): The first interpretation is a lexical common noun representing a month of the year, as used in the following sentence:

   (59) a. Bu işi Ekim'de bitirmeliydik.
   `this job October+LOC finish+NECS+PAST+1PL`
   'We should have finished this job in October.'

   Regarding this interpretation the system produces the following feature structure:

$$
{}_{\textit{lexical common}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common}\end{bmatrix} \\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“ekim”} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \\ \text{CASE} & \text{loc}\end{bmatrix} \\
\text{SYN} & \begin{bmatrix}\ldots\end{bmatrix} \\
\text{SEM} & \begin{bmatrix}\text{TEMPORAL} & + \\ \ldots & \end{bmatrix} \\
\text{PHON} & \text{“ekimde”}
\end{bmatrix}
$$

   The query form subsumes the structure above, hence it satisfies the restrictions.

2. *ekimde* (*in my appendix/suffix*): The second interpretation is also a lexical common noun, for which there are two senses in the static lexicon: *appendix* and *suffix*. Feature structures for both of the senses are similar, so we will consider only the first one, *appendix*, which is used in the following sentence:

   (60) a. O şekil benim ekimde olmalıydı.
   `that figure my appendix+P1SG+LOC be+NECS+PAST+3SG`
   'That figure should have been in my appendix.'

   The full feature structure for the second interpretation, *in my appendix*, is the following:

$$
{}_{\textit{lexical common}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“ek”} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{1sg} \\ \text{CASE} & \text{loc} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \ldots \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{TEMPORAL} & - \\ \ldots & \end{bmatrix} \\
\text{PHON} & \text{“ekimde”}
\end{bmatrix}
$$

Due to the $-$ value of SEM | TEMPORAL and *1sg* value of MORPH | POSS features, the subsumption of the feature structure above with the query form will fail, and it will be eliminated. Note that both of the restriction features are appropriate for the feature structures above.

4.2 Producing Feature Structures

We will describe the processing in the lexicon as consisting of three main steps:

1. morphological analysis,
2. retrieval of syntactic and semantic information and unification with morphosyntactic information,
3. application of restrictions.

The first step is external to the system, so we will consider only its input/output interface. The second step consists of transformation of morphological parses to feature structure syntax, category mapping, retrieval from static lexicon, and computing features according to the morphological parses. The final step is relatively simple; it just tests the sumbsumtion of input query form with each of the produced structures.

In the next sections, we will examine each step and provide details with examples.

4.2.1 Morphological Analysis

Morphological processor provides possible interpretations of a surface form. Due to the rich set of inflectional and derivational suffixes in Turkish, it is highly probable that the surface form will have more than one interpretation. Consider the possible interpretations of the surface form *kazma*, for which the morphological processor output is given in Figure 4.3, as used in the following examples:

(61) a. Dün burada bir kazma gördün mü?
`yesterday here a pickaxe see+PAST+2SG QUES`
'Did you see a pickaxe here yesterday?'

b. Orayı sakın kazma!
`there never dig+NEG+2SG`
'Do not dig there!'

c. Kazma işini sanırım bugün
`dig+INF job+P3SG+ACC guess+ARST+1SG today`
bitiririz.
`finish+ARST+1PL`
'I guess we will finish digging today.'

```
1. [[CAT=NOUN][ROOT=kazma][AGR=3SG][POSS=NONE][CASE=NOM]]
2. [[CAT=VERB][ROOT=kaz][SENSE=NEG][TAM1=IMP][AGR=2SG]]
3. [[CAT=VERB][ROOT=kaz][SENSE=POS]
    [CONV=NOUN=MA][TYPE=INFINITIVE][AGR=3SG][POSS=NONE][CASE=NOM]]
```

Figure 4.3: Interpretations of the surface form *kazma*.

The first interpretation contains the noun reading, *pickaxe*. The second and third interpretations consider the verb *kaz-* (*dig*). In the second interpretation, the suffix *ma* is an inflectional suffix and negates the predicate, as opposed to the other one, which is a derivational suffix and used to derive the infinitive *kazma* (*digging*).

As seen in the example above, the rich set of inflectional and derivational suffixes causes many interpretations, which increase in number when the multiple senses are incorporated. For example, the predicative verb *ye* has at least four senses, which we mentioned in Section 3.7.1.

The morphological processor output must be transformed to feature structure syntax, moreover, due to the comprehensive categorization introduced in Chapter 3, category mapping will take place. The following section describes this transformation and retrieving information in the static lexicon.

4.2.2 Retrieving Information in the Static Lexicon

The static lexicon follows the interpretations produced by the morphological processor. Interpretations include category information, the root words, and a number of inflectional and derivational suffixes, such as case and possessive markers. The retrieval step mainly consists of the following phases:

- transformation of interpretations into feature structure syntax, and correct mapping from the morphological processor category to the static lexicon category,
- accessing the feature structures of the root words involved in the morphological parses, and computing features accordingly.

During the processing, the system accesses two tables and two databases. The tables are used to map category information, and the databases are used to access feature structures of the root words containing syntactic and semantic information (i.e., lexical database), and the template structures.

The retrieval process starts with transformation of parses into feature structure syntax, since the syntactic and semantic information is stored in the form of feature structures in the static lexicon. As seen in the interpretations of *kazma* in the previous section, derivations exist in morphological parses and may go to arbitrary depth, such as *Çekoslovakyalılaştıramadıklarımızdanmışsınız*.

As another example for the interpretations containing derivations, consider the one in Figure 4.4. It starts with the noun akıl (*intelligence*), which is used to derive the adjective *akıllı* (*intelligent*). The derivations end with the manner adverb *akıllıca* (*intelligently*). The derivations in the processor output are highlighted with the CONV item in the string below, which gives the category and derivational suffix. Thus, in the following example, there are two derivations and three categories traversed, that is there are three levels: the first is the lexical level and the other two are the derivational levels. Each level is transformed into a feature structure containing category and morphosyntactic information. So, the interpretation above would be transformed into a list of levels with three elements.

```
[[CAT=NOUN][ROOT=akIl][CONV=ADJ=LI][CONV=ADVERB=CA][TYPE=MANNER]]
```

Figure 4.4: The derivation path to the manner adverb *akıllıca*.

While transforming the interpretations, the system maps the category information in the morphological processor output to correct lexicon category for all levels, which is due to the finer-grained categorization of the lexicon. For this purpose, two tables are maintained for root words and derivations, respectively. For the first one, processor category and root word uniquely determine the lexicon category. For each root word represented in the feature structure database, an entry in this table must be present. A portion of such a table for nouns is depicted in Figure 4.5. For the second table, processor category and derivational suffix uniquely determine the lexicon category. This mapping is given in Table 4.1.

Processor category	Root word	Lexicon category
...	...	...
noun	kazıntı	common noun
noun	**kazma**	**common noun**
noun	kazmanoğlu	proper noun
noun	ketçap	common noun
noun	...	...
noun	kurtuluş	proper noun
...	...	...

Figure 4.5: A portion of the table used for category mapping for root words.

This step is applied to all of the morphological parses, and at the end of this step, for each parse there is a list of levels, each of which contains the correct lexicon category and a set of features representing morphosyntactic information of interpretations.

The next phase in the processing is the retrieval of the syntactic and semantic information and producing feature structures. The syntactic and semantic information about the root words is stored in the feature structure database, which is indexed with the category and the root word information. For the root words in the lexical levels of each parse, the feature structure database is accessed and matching entries are retrieved. However, the entries contain only syntactic and semantic information for the non-derived forms, thus morphosyntactic information needs to be unified and by following the derivation information of parses new feature structures should be constructured. Many examples of this phenomenon are presented in the Chapter 3.

Since the morphological parses are previously transformed into feature structure syntax, unification of morphosyntactic information is simple. Having unified all the information, the processing for the lexical level is completed. If the morphological parses do not contain a derivation to another category, the process above is sufficient to produce the result. However, as we have already mentioned, the cases in which derivations exist are not rare.

For each derivation in the parses, a new feature structure is constructed. For this purpose, using the category information in the derivational levels, the template feature structure database is accessed and corresponding template feature structures are retrieved. These structures do not contain feature values, but they will be computed by the system.

Morphological Processor Output		Lexicon Category				
Category	Suffix	MAJ	MIN	SUB	SSUB	SSSUB
noun	cı, lık, cık, og, yıcı, mazlık, yamazlık, maca, yası, *none*	nominal	noun	common		
	mak		sentential	act	infinitive	mak
	ma					ma
	yış					yış
	dık			fact	participle	dık
	yacak					yacak
rpronoun	*none*	noinal	pronoun	quantitative		
adj	lık, lı, ki, sız, sı, ik, yıcı, yan, yacak, dık, yası	modifier	adjective	qualitative		
adverb	yınca, yıp	adverbial	temporal	point-of-time		
	yalı, ken			time-period	fuzzy	
	casına, maksızın, madan, yamadan, yerek, ca		manner	qualitative		
	dıkça			repetition		
verb	lan, laş	verb	predicative			
	none	verb	attributive			

Table 4.1: The table used for category mapping for derived words.

Starting from the leftmost derivational level, the derivation path is followed: for each derivation a new feature structure is constructed; feature values are computed. The result is a nested feature structure, in which the previous structures are stored in MORPH | STEM feature as shown in Figure 4.6.

Having retrieved the template feature structure, the feature values are to be computed by the system. Morphosyntactic information is already produced by the morphological processor, and unified with the information in the template structures. A feature structure belonging to any category should has the following minimum information: category, phonology, stem, concept, and form. Among them the category information and the form (i.e., it is *derived*) are already known. The feature MORPH | STEM holds the feature structures of the previous words, as described above. The phonology information is valid only in the last feature structure in the derivation, whose value is the surface form given as the input to the morphological processor.[1] The concept feature is computed by means of a function according to the target derivation category and suffix.

[1] In other structures, this value is undefined, although computation is possible by means of morphological generation.

$$
{}_{\textit{derived}\ \ldots}\begin{bmatrix}
\text{CAT} & \ldots \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & {}_{\textit{derived}\ \ldots}\begin{bmatrix} \text{CAT} & \ldots \\ \text{MORPH} & \begin{bmatrix} \text{STEM} & \begin{bmatrix}\ldots\end{bmatrix} \\ \text{FORM} & \text{derived} \\ \text{CASE} & \ldots. \\ \ldots & \end{bmatrix} \\ \text{SYN} & \ldots \\ \text{SEM} & \ldots \end{bmatrix} \\ \text{FORM} & \text{derived} \\ \text{CASE} & \ldots \\ \ldots & \end{bmatrix} \\
\text{SYN} & \ldots \\
\text{SEM} & \ldots \\
\text{PHON} & \ldots
\end{bmatrix}
$$

Figure 4.6: Nested feature structures.

There are other features to be computed other than the common ones, among which subcategorization information and thematic roles are the most important ones. These are co-indexed with the those of the previous derivational level. Furthermore, a number of features specific to some categories exist, e.g., semantic properties of common nouns or the constraints on the modified of qualitative adjectives. About the second one, for example, the following prediction can be made: qualitative adjectives modify the common nouns, and do not constrain the agreement and countability features. However, predicting the semantic properties is difficult, and for this reason, the default values are used, which may not always give the correct description.

In the next section we will clarify the procedure above by giving examples.

Examples

In summary, the process of producing feature structures follows the following steps:

1: For each parse in the morphological processor output do the following:

1.1: Find the lexicon category of the initial root word (see the table in Figure 4.5),

1.2: Find the lexicon entries of all senses of the root word by matching the root word information,

1.3: Unify morphosyntactic information with the information in the lexicon entry/entries,

1.4: While there is derivation in the parse do the following:

1.4.1: Find the lexicon category and retrieve the corresponding template feature structure (see Table 4.1),

1.4.2: Compute feature values and unify morphosyntactic information,

1.5: Output the feature structure(s)

We will describe the process with the input surface form *kazma*, which has three interpretations, one of which includes a derivation (see example (61) and Figure 4.3 for morphological processor output):

1. *Kazma* (common noun): This interpretation is due to the common noun *kazma* (*pickaxe*), and does not contain a derivation, so the result can be easily produced by combining morphosyntactic, syntactic, and semantic information.

   As we already described, the process starts with determining the lexicon category. The morphological processor categorizes *kazma* just as a noun, however, it is represented as a *common noun* in the static lexicon. Then, the corresponding feature structure in the lexicon is searched by matching the ROOT information of morphological processor with MORPH | STEM feature of lexicon entries. The matching feature structure is given below. Note that there is only one sense of *kazma* (*pickaxe*) in our lexicon.

$$
{}_{\textit{lexical common}}\left[\begin{array}{ll}
\text{CAT} & \begin{bmatrix}\text{MAJ} & \text{nominal}\\ \text{MIN} & \text{noun}\\ \text{SUB} & \text{common}\end{bmatrix}\\
\text{MORPH} & \begin{bmatrix}\text{STEM} & \text{“kazma”}\\ \text{FORM} & \text{lexical}\end{bmatrix}\\
\text{SYN} & \begin{bmatrix}\text{SUBCAT} & \text{none}\end{bmatrix}\\
\text{SEM} & \begin{bmatrix}\text{CONCEPT} & \text{\#kazma-(pickaxe)}\\ \text{COUNTABLE} & +\end{bmatrix}
\end{array}\right]
$$

   Then, information about inflectional suffixes are unified with the lexicon entry, which produces the result:

$$
{}_{\textit{lexical common}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{noun} \\ \text{SUB} & \text{common} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“kazma”} \\ \text{FORM} & \text{lexical} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3pl} \\ \text{POSS} & \text{1sg} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \text{SUBCAT} & \text{none} \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#kazma-(pickaxe)} \\ \text{COUNTABLE} & + \end{bmatrix} \\
\text{PHON} & \text{“kazma”}
\end{bmatrix}
$$

Note that the phonology information is the same as surface form given as an input to the system.

2. *Kazma* (verb): This interpretation comes from the verbal root *kaz-* (*dig*). The suffix *ma* is an inflectional suffix, which negates the meaning (see Figure 4.3 for the parse). Since no derivation step is involved, the process is similar to that of the common noun reading. The lexicon entry is given below with the morphosyntactic information unified:

$$
{}_{\textit{lexical predicative verb}}\begin{bmatrix}
\text{CAT} & \begin{bmatrix} \text{MAJ} & \text{verb} \\ \text{MIN} & \text{predicative} \end{bmatrix} \\
\text{MORPH} & \begin{bmatrix} \text{STEM} & \text{“kaz”} \\ \text{FORM} & \text{lexical} \\ \text{SENSE} & \text{neg} \\ \text{TAM1} & \text{imp} \\ \text{AGR} & \text{2sg} \end{bmatrix} \\
\text{SYN} & \begin{bmatrix} \text{SUBCAT} & \ldots \end{bmatrix} \\
\text{SEM} & \begin{bmatrix} \text{CONCEPT} & \text{\#kaz-(to dig)} \\ \text{ROLES} & \ldots \end{bmatrix} \\
\text{PHON} & \text{“kazma”}
\end{bmatrix}
$$

3. *Kazma* (infinitive): This interpretation involves a derivation from the verb *kaz-* (*dig*) to the infinitive *kazma* (*digging*). The steps up to the derivation is similar to that of the previous two examples. The derivation step starts with the determination of the target category using the Table 4.1, and retrieval of the template feature structure. The table

lookup results in the *infinitive* category, and corresponding template feature structure is retrieved.

The next step involves the computation of features, which includes subcategorization information, thematic roles, and concept. These features, except the concept, are co-indexed with the corresponding entries in the lexicon entry of *kaz-*. The concept feature is computed via a function. The rest of the features can be easily found, since category is already known and morphosyntactic information is received from the morphological processor. The phonology feature takes the input surface form, *kazma*.

The feature structure for the infinitive *kazma* is given below, with some of the features co-indexed with those of the lexical entry of *kaz-*:

$$
{}_{ma}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{nominal} \\ \text{MIN} & \text{derived} \\ \text{SUB} & \text{act} \\ \text{SSUB} & \text{infinitive} \\ \text{SSSUB} & \text{ma} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \boxed{1} \\ \text{DERV-SUFFIX} & \text{“ma”} \\ \text{FORM} & \text{derived} \\ \text{CASE} & \text{nom} \\ \text{AGR} & \text{3sg} \\ \text{POSS} & \text{none} \end{array}\right] \\
\text{SYN} & \left[\begin{array}{ll} \text{SUBCAT} & \boxed{2} \end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \text{f}_{ma}(\text{\#kaz-(dig)}) \\ \text{ROLES} & \boxed{3} \end{array}\right] \\
\text{PHON} & \text{“kazma”}
\end{array}\right]
$$

$$
\boxed{1}\ {}_{\textit{lexical predicative verb}}\left[\begin{array}{ll}
\text{CAT} & \left[\begin{array}{ll} \text{MAJ} & \text{verb} \\ \text{MIN} & \text{predicative} \end{array}\right] \\
\text{MORPH} & \left[\begin{array}{ll} \text{STEM} & \text{“kaz”} \\ \text{FORM} & \text{lexical} \\ \text{SENSE} & \text{pos} \end{array}\right] \\
\text{SYN} & \left[\begin{array}{ll} \text{SUBCAT} & \boxed{2} \ldots \end{array}\right] \\
\text{SEM} & \left[\begin{array}{ll} \text{CONCEPT} & \text{\#kaz-(dig)} \\ \text{ROLES} & \boxed{3} \ldots \end{array}\right] \\
\text{PHON} & \text{none}
\end{array}\right]
$$

4.2.3 Application of Restrictions

The final step in the process is the elimination of the feature structures that do not satisfy the restrictions.

The input to this phase is a list of feature structures and the user's query form. Each structure is tested against the query form for subsumtion relation, that is all of the features in the query form must be present in the output structures and the feature values must be the same. The ones that fail to satisfy this relation are eliminated.

The process is relatively simple, thus we will not decribe it any further (see the example in Section 4.1).

4.3 Problems and Limitations

A limitation with the representation of the entries in the static lexicon is related with the SEM | CONCEPT feature, which gives a brief English description of the object, event, etc. that the root word represents. The description is mostly human-oriented and not directly usable by NLP subsystems, such as transfer units (from Turkish to English and vice versa) in machine translation systems. For example, this feature may take the value *throw a physical object* for the verb *at-*. Using an ontological component in the lexicon eliminates this problem, in which concepts would be described via a semantic network.

Another problem that the ontological component would eliminate is the following: the subcategorization information for verbs, common nouns, etc. may places some semantic constraints on the complements, such as the agent of the verb *ye-* (*eat something*) must be animate (SEM | ANIMATE is +). This constraint would be tested with the semantic feature in the feature structure of the subject during syntactic analysis. This test, however, may fail due to the absence of the feature SEM | ANIMATE, but this structure may describe a human, such as *öğrenci student* having SEM | HUMAN:+, so satisfying animateness constraint. This syntactic mismatch of the features would be eliminated easily, since a human object would inherit animateness property (see Yılmaz [16] for such a component in a verb lexicon).

One of the problems with producing feature structures, especially with the derivations involved, is predicting semantic properties of common nouns and qualitative adjectives. In the other categories either semantic properties are not introduced or they do not receive derivation.

Since the new word generated as a result of the derivation process does not have a lexicon entry, the process should predict some feature values. However, the semantics of the object or the quality that the derivation process produces is not clear. For example, consider the derivation that takes a common noun and the suffix *cı*, and produces a common noun. Both *akşamcı* and *öğlenci* are produced in this way, however, the semantic properties of the resultant entities are not predictable. This is the case in *yazıcı* (*yaz-* (*write*)+*cı*), which has two senses: *printer* and *the person who writes*. The two senses have different properties, e.g., animateness.

A similar situation occurs for the qualitative adjectives. For instance, as we stated previously, the gradability of derived forms are not quite predictable: *çok akılsız* vs. **çok kolsuz*.

Chapter 5

Implementation

The processing in the lexicon consists of four main steps each carried out by a separate module:

1. morphological analysis,
2. transformation of morphological processor output to static lexicon the syntax (i.e., feature structure syntax), and category mapping,
3. retrieval from feature structure databases and producing feature structures,
4. application of restrictions.

Except the morphological processor component,[1] which is previously implemented, all the components are implemented in *SICStus Prolog release 3 #5* [14]. Since we described the procedural aspects of the lexicon in Chapter 4, we will not go into the details of this process, however, there is one point to be made here: in the implementation, the query form can contain features only from CAT and MORPH, since the lexicon interface does not gain much by adding the capability of restricting SYN and SEM features, as well. On the other hand, NLP subsystems using this interface can impose any restriction externally, because access to all features is allowed. So, rather than applying restrictions to eliminate unwanted feature structures as the final step, the system applies restrictions to *parses* right after the transformation phase (i.e., when the CAT and MORPH features are computed). Thus, unnecessary retrievals and computations are avoided.

We provided a procedural interface for the lexicon, rather than implementing a graphical one, since the interface will be open to NLP subsystems in practical applications.

[1] The morphological processor that our lexicon employs is implemented by Oflazer (see Oflazer [11] for the two level description of Turkish morphology) using a finite-state lexicon compiler by Karttunen [7].

In this chapter, we will first describe an important component of the system, the feature structure database (i.e., the root word lexicon). Then, we will give outputs from sample runs of the system.

5.1 Feature Structure Database

The feature structure database consists of a list of feature structures indexed with category and root word. Each word and sense is a separate entry in the database, so given a category and root word more than one entry may match, that is the key is not unique. Each entry is a unit Prolog clause with seven arguments, the first five ones giving the category, and the other two giving the root word and the corresponding feature structure (see Figure 5.1). In this way, the database can be stored in the main memory and allows fast access.

```
fsdb(verb, existential, none, none, none, var,
     [cat:[maj:verb, ...], syn:[...], ...]).
```

Figure 5.1: The entry for the existential verb *var* in the feature structure database.

Feature structures are represented as a list of <*feature name*:*feature value*> pairs (see Gazdar and Mellish [4]). For example, the following feature structure with abstract representation would be represented in Prolog as in Figure 5.2:

$$\begin{bmatrix} \text{MORPH} & \begin{bmatrix} \text{STEM} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{MAJ} & \text{nominal} \end{bmatrix} \end{bmatrix} \\ \text{CASE} & \text{dat} \end{bmatrix} \\ \text{SEM} & \begin{bmatrix} \text{ANIMATE} & - \\ \text{COUNTABLE} & - \end{bmatrix} \end{bmatrix}$$

```
[morph:[stem:[cat:[maj:nominal |_] |_], case:dat |_],
        sem:[animate:-, countable:- |_] |_]
```

Figure 5.2: Prolog representation of a feature structure.

Currently, our feature structure database contains about 50 entries, which consists of samples from the closed-class words, such as post-positions, conjunctions, and from other categories showing some special property. More entries will be added to the system later. In order to maintain the database, the system provides a number of predicates to add, delete, and browse entries.

5.2 Sample Runs

In this section we will present three sample runs that will demonstrate features of our lexicon, and will clarify the algorithms presented in Chapter 4.

The input to the system is a query form in the form of a feature structure. At least the PHON feature, which holds the surface form, must be present in the query form. Other features are optional, and if present they act as restrictions on the final output feature structures. The user can test presence of a feature or a specific value for that feature. If the feature restricted is in the output feature structure, the restriction value, which may be unspecified to test the presence, is unified with the one in the output structure. If the unification fails, the output structure is eliminated. If such a feature is not in the output structure, the restriction feature would not be appropriate for this structure, so it is again eliminated; for example MORPH | TAM1 feature is not appropriate for a conjunction's feature structure.

As previously mentioned, the process is divided into four phases in the implementation. All four phases inform the user about the state of the processing. The final output is a list of feature structures which satisfy all the constraints.

5.2.1 Example 1

The first example submits only the surface form *atım* and does not constrain any other features. According to the morphological processor, *atım* has three parses, as illustrated by the following examples:

(62) a. Benim bir atım var.
`my a horse+P1SG existent`
'I have a horse.'

b. Küheylan ben bir atım dedi.
`Küheylan I a horse+PRES+1SG say+PAST+3SG`
'Küheylan said that it was a horse.'

c. Tilki bir atım mesafedeydi.
`fox one shot distance+PAST+3SG`
'The fox was in one shot distance.'

The category of the surface form *atım* is common noun and attributive verb, respectively, in the first two parses, and they are due to the common noun *at* (*horse*). The third parse comes from the common noun *atım* (*shot*), and does not derive to another category. Since query form does not place any constraint, the system will generate output for all of the parses, as far as the feature structure database contains corresponding entries.

The user input and the lexicon's output follow:

Input query form:[2]

```
[phon:atIm]
```

Output:

```
Parsing surface form started...

     Reading Turkish binary file...
     0%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>100%
     Read Turkish binary file.

     Parsing: atIm
     Number of parses: 3
     1: [[CAT=NOUN][ROOT=at][AGR=3SG][POSS=1SG][CASE=NOM]]
     2: [[CAT=NOUN][ROOT=at][AGR=3SG][POSS=NONE][CASE=NOM]
         [CONV=VERB=NONE][TAM2=PRES][AGR=1SG]]
     3: [[CAT=NOUN][ROOT=atIm][AGR=3SG][POSS=NONE][CASE=NOM]]

Parsing surface form ended...

Transformation phase started...

     Category mapping from:
          noun, none and at
     to:
          nominal, noun, common, none, none

     Category mapping from:
          noun, none and at
     to:
          nominal, noun, common, none, none

     Category mapping from:
```

[2] In our system, Turkish words consist of all lowercase letters, and ı, ç, ğ, ş, ö, and ü are represented as the capital of the nearest letter.

```
          verb, none and none
     to:
          verb, attributive, none, none, none

     Exception: Entry not found in LCMT: Skipping parse...
          noun
          none
          atIm

Transformation phase ended...

     Transformed parses:
     -------------------
     Parse information:
          Number of parses: 2
          1: 1 level(s)
          2: 2 level(s)

Application of restrictions phase started...

Application of restrictions phase ended...

     Satisfying parses:
     ------------------
     Parse information:
          Number of parses: 2
          1: 1 level(s)
          2: 2 level(s)

Retrieval phase started...

     Access to FSDB with:
          nominal, noun, common, none, none and at
     for:
          1 entry/entries

     Access to FSDB with:
          nominal, noun, common, none, none and at
     for:
          1 entry/entries

     Access to TFSDB with:
```

```
          verb, attributive, none, none, none

Retrieval phase ended...

     Final result:
     -------------
     Number of feature structures: 2
     Feature sturucture(s):

        [sem:
              [countable: +
               animate: +
               concept: at-(horse)
               material: -
               unit: -
               container: -
               spatial: -
               temporal: -]
         cat:
              [maj: nominal
               min: noun
               sub: common
               ssub: none
               sssub: none]
         morph:
              [stem: at
               form: lexical
               case: nom
               poss: 1sg
               agr: 3sg]
         syn:
              [subcat: none]
         phon: atIm]
         ,
         [cat:
              [maj: verb
               min: attributive
               sub: none
               ssub: none
               sssub: none]
         morph:
              [stem:
```

```
                [sem:
                     [countable: +
                      animate: +
                      concept: at-(horse)
                      material: -
                      unit: -
                      container: -
                      spatial: -
                      temporal: -]
                 cat:
                     [maj: nominal
                      min: noun
                      sub: common
                      ssub: none
                      sssub: none]
                 morph:
                     [stem: at
                      form: lexical
                      case: nom
                      poss: none
                      agr: 3sg]
                 syn:
                     [subcat: none]
                 phon: none]
            form: derived
            derv_suffix: none
            tam2: pres
            copula: none
            agr: 1sg]
      syn:
          [subcat: none]
      sem:
          [concept: none(at-(horse))
           roles: none]
      phon: atIm]
```

The output is a trace of the four phases. The first part is the morphological parsing, and displays parses. The second part is the transformation of parses into static lexicon syntax (i.e., feature structure syntax), and category mapping. The first item in the output of this phase shows the mapping of the morphological processor category *noun* to the lexicon category *common noun* for the root word *at*. The next two output items illustrate category mapping of the second parse. The last item shows that the category mapping table for root words does not have an entry for *atım*, that is the system does not have information about *atım*, so this parse is omitted, and will not be processed in the following phases.

After the transformation phase, two parses remain, and since no restriction is imposed by the user, these parses will pass to the next phase. The retrieval part acknowledges the user that it accessed the feature structure database entry of the common noun *at* two times, and the template feature structure for attributive verbs, which is due to the derivation in the second parse.

Each parse produces only one feature structure, because the common noun *at* has only one entry/sense in the database. The final output is these feature structures. The processing including interfacing with the morphological processor, producing feature structures, and pretty-printing takes approximately 30 msec. of running time for compiled Prolog code, so it is rather fast. As we mentioned in Chapter 2, the number of lexical items in a lexicon of a system with acceptable coverage (e.g., The Core Language Engine) will not exceed a few thousand, so whole database can be stored in the main memory. Thus, as the size of our lexical database gets larger, the processing time will not exceed acceptable limits.

5.2.2 Example 2

This example run submits the surface form *memnunum* to the system and constraints the output to be of category *verb*. Given this surface form, morphological processor gives three parses as used in the following examples:[3]

(63) a. Senden memnunum.
`you+GEN happy+PRES+1SG`
'I am happy with you.'

b. Memnunum benim!
`happy one+P1SG my`

c. Ben Memnun'um.
`I Memnun+PRES+1SG`
'I am Memnun.'

The first two parses are due to the qualitative adjective *memnun* (*satisfied*/*happy*), and contain derivations to attributive verb and common noun, respectively. The last one is due to the

[3] The usage in the second sentence is like in *güzelim benim*, that is the qualitative adjective *güzel* (*beautiful*) is subject to a derivation to common noun, and becomes *the one that is beautiful*. This usage of *Memnun* is syntacticly correct, though semantically it does not make sense.

proper noun *Memnun* and contains a derivation to attributive verb. The only restriction in the query form is that the output feature structures must be of type *verb*, which will cause the second parse to be eliminated in the third phase.

The input and corresponding output follow:

Input query form:

```
[phon:memnunum, cat:[maj:verb]]
```

Output:

```
Parsing surface form started...

     Parsing: memnunum
     Number of parses: 3
     1: [[CAT=ADJ][ROOT=memnun][CONV=VERB=NONE][TAM2=PRES][AGR=1SG]]
     2: [[CAT=ADJ][ROOT=memnun][CONV=NOUN=NONE][AGR=3SG][POSS=1SG][CASE=NOM]]
     3: [[CAT=NOUN][ROOT=memnun][TYPE=RPROPER][AGR=3SG][POSS=NONE][CASE=NOM]
         [CONV=VERB=NONE][TAM2=PRES][AGR=1SG]]

Parsing surface form ended...

Transformation phase started...

     Category mapping from:
          adj, none and memnun
     to:
          adjectival, adjective, qualitative, none, none

     Category mapping from:
          verb, none and none
     to:
          verb, attributive, none, none, none

     Category mapping from:
          adj, none and memnun
     to:
          adjectival, adjective, qualitative, none, none

     Category mapping from:
          noun, none and none
```

```
     to:
          nominal, noun, common, none, none

     Exception: Entry not found in LCMT: Skipping parse...
          noun
          rproper
          memnun

Transformation phase ended...

     Transformed parses:
     -------------------
     Parse information:
          Number of parses: 2
          1: 2 level(s)
          2: 2 level(s)

Application of restrictions phase started...

     Parse eliminated: Printing only the last level...

         [cat:
              [maj: nominal
               min: noun
               sub: common
               ssub: none
               sssub: none]
          morph:
              [derv_suffix: none
               agr: 3sg
               poss: 1sg
               case: nom]
          phon: memnunum]

Application of restrictions phase ended...

     Satisfying parses:
     ------------------
     Parse information:
          Number of parses: 1
          1: 2 level(s)
```

```
Retrieval phase started...

     Access to FSDB with:
          adjectival, adjective, qualitative, none, none and memnun
     for:
          1 entry/entries

     Access to TFSDB with:
          verb, attributive, none, none, none

Retrieval phase ended...

     Final result:
     -------------
     Number of feature structures: 1
     Feature sturucture(s):

         [cat:
              [maj: verb
               min: attributive
               sub: none
               ssub: none
               sssub: none]
          morph:
              [stem:
                   [syn:
                        [subcat: ...
                         modifies: ...]
                    cat:
                        [maj: adjectival
                         min: adjective
                         sub: qualitative
                         ssub: none
                         sssub: none]
                    morph:
                        [stem: memnun
                         form: lexical]
                    sem:
                        [concept: memnun-(satisfied)
                         gradable: -
                         questional: -]
                    phon: none]
```

```
          form: derived
          derv_suffix: none
          tam2: pres
          copula: none
          agr: 1sg]
     syn:
         [subcat: ...]
     sem:
         [concept: none(memnun-(satisfied))
          roles: none]
     phon: memnunum]
```

In the transformation of parses, no entry regarding the proper noun *Memnun* is found in the category mapping table, so this parse is eliminated, leaving two parses to the third phase, which discards the second parse, since it fails to satisfy the restriction, that is the value of CAT | MAJ must be *verb*. Finally, there is only one parse left, which is the first one, as an input to the retrieval phase. As seen in the output, there is only one entry for the qualitative adjective *memnun*, thus only one feature structure is generated. The processing takes approximately 50 msec. of running time. The values of SUBCAT and MODIFIES features are omitted to save space (see the full feature structure of *memnun* on page 40).

5.2.3 Example 3

Our last example will demonstrate multiple senses in the database. The surface form is *ekim*, and the restriction is on MORPH | POSS feature, whose value must be *1sg*. The interpretations are similar to those in the previous examples, so we will not give detailed descriptions.

According to the morphological processor, there are three parses, which are due to the common noun *ek* (*appendix/suffix*) and *Ekim* (*October*). Both root words are in the database, but the last two parses are eliminated in the third phase. As a result, there is only one parse as an input to the last step. There are two entries regarding the common noun *ek*, which cause the system to generate two feature structures for the single parse. The processing takes about 40 msec.

The input and corresponding output follow:

Input query form:

```
[phon:ekim, morph:[poss:'1sg']].
```

Output:

```
Parsing surface form started...

     Parsing: ekim
     Number of parses: 3
     1: [[CAT=NOUN][ROOT=eK][AGR=3SG][POSS=1SG][CASE=NOM]]
     2: [[CAT=NOUN][ROOT=eK][AGR=3SG][POSS=NONE][CASE=NOM]
         [CONV=VERB=NONE][TAM2=PRES][AGR=1SG]]
     3: [[CAT=NOUN][ROOT=ekim][TYPE=TEMP1][AGR=3SG][POSS=NONE][CASE=NOM]]

Parsing surface form ended...

Transformation phase started...

     Category mapping from:
          noun, none and ek
     to:
          nominal, noun, common, none, none

     Category mapping from:
          noun, none and ek
     to:
          nominal, noun, common, none, none

     Category mapping from:
          verb, none and none
     to:
          verb, attributive, none, none, none

     Category mapping from:
          noun, temp1 and ekim
     to:
          nominal, noun, common, none, none

Transformation phase ended...

     Transformed parses:
     -------------------
     Parse information:
          Number of parses: 3
          1: 1 level(s)
```

```
          2: 2 level(s)
          3: 1 level(s)

Application of restrictions phase started...

     Parse eliminated: Printing only the last level...

         [cat:
              [maj: verb
               min: attributive
               sub: none
               ssub: none
               sssub: none]
          morph:
              [suffix: none
               tam2: pres
               agr: 1sg]
          phon: ekim]

     Parse eliminated: Printing only the last level...

         [cat:
              [maj: nominal
               min: noun
               sub: common
               ssub: none
               sssub: none]
          morph:
              [stem: ekim
               agr: 3sg
               poss: none
               case: nom]
          phon: ekim]

Application of restrictions phase ended...

     Satisfying parses:
     ------------------
     Parse information:
          Number of parses: 1
          1: 1 level(s)
```

```
Retrieval phase started...

     Access to FSDB with:
          nominal, noun, common, none, none and ek
     for:
          2 entry/entries

Retrieval phase ended...

     Final result:
     -------------
     Number of feature structures: 2
     Feature sturucture(s):

         [sem:
              [countable: +
               concept: ek-(suffix)
               material: -
               unit: -
               container: -
               spatial: -
               temporal: -
               animate: -]
          cat:
              [maj: nominal
               min: noun
               sub: common
               ssub: none
               sssub: none]
          morph:
              [stem: ek
               form: lexical
               case: nom
               poss: 1sg
               agr: 3sg]
          syn:
              [subcat: none]
          phon: ekim]
         ,
         [sem:
              [countable: +
               concept: ek-(appendix)
```

```
      material: -
      unit: -
      container: -
      spatial: -
      temporal: -
      animate: -]
cat:
     [maj: nominal
      min: noun
      sub: common
      ssub: none
      sssub: none]
morph:
     [stem: ek
      form: lexical
      case: nom
      poss: 1sg
      agr: 3sg]
syn:
     [subcat: none]
phon: ekim]
```

Chapter 6

Conclusions and Suggestions

In this thesis, we present a lexicon for Turkish. Our work includes determination of the lexical specification to be encoded for all lexical types of Turkish, encoding of this specification, and constructing a standalone system as an information repository for the NLP systems.

The level of lexical specification for morphosyntactic and syntactic information is adequate, but, as the semantic information is added in an ad hoc manner, it may not satisfy all the requirements of NLP systems on semantic information. Including a knowledge-base/ontology into the system, in which concepts are described through a semantic network, would be useful. This would solve the problem related with the satisfying the semantic constraints in the subcategorization information of lexical entries. For example, the constraint posing SEM | ANIMATE:+ will not be unified with SEM | HUMAN:+, though this is semantically satisfiable.

In order for our lexical database to be computationally useful, more entries would be added depending on the requirements of the NLP systems interfacing with our lexicon. Currently, the database contains about 50 entries consisting of samples from closed and open-class words having some special property. We are planning to add more entries to cover all the closed-class words and enrich the content for the open-class words of Turkish. A graphical user interface will be provided to help insertion, deletetion, and update operations to lexicon.

Bibliography

[1] T. Briscoe. *Natural Language and Speech*, chapter Lexical Issues in Natural Language Processing. Springer-Verlag, 1991.

[2] D. Carter. *The Core Language Engine*, chapter Lexical Acquisition. The MIT Press, 1992.

[3] H. Ediskun. *Türk Dilbilgisi*. Remzi Kitabevi, İstanbul, 4. edition, 1993.

[4] G. Gazdar and C. Mellish. *Natural Language Processing in Prolog*, chapter 7, pages 217–278. Addison-Wesley Publishing Company, 1989.

[5] K. Goodman and S. Nirenburg. *KBMT-89: A Case Study in Knowledge-Based Machine Translation*. San Mateo: Morgan Kaufmann, 1991.

[6] R. Grishman and N. Calzolari. Lexicons. In *Survey of the State of the Art in Human Language Technology*, chapter Language Resources. Available at http://www.cse.ogi.edu/CSLU/HLTsurvey/.

[7] L. Karttunen. Finite-state lexicon compiler. Technical report, XEROX Palo Alto Research Center, April 1993.

[8] N. Koç. *Yeni Dilbilgisi*. İnkılap Kitabevi, İstanbul, 1. edition, 1990.

[9] M. Monachini and N. Calzolari. Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Applicatios to European Languages. Draft version, October 1994.

[10] S. Nirenburg. Lexicon Acquisition for NLP: A Consumer Report. Technical report, Carnegie Mellon University, Center for Machine Translation, 1989.

[11] K. Oflazer. Two-level Description of Turkish Morphology. *Literary and Linguistic Computing*, 9(2), 1994. Oxford University Press.

[12] S. M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*. Stanford University, Center for the Study of Language and Information, 1986. CSLI Lecture Notes.

[13] A. Solak and K. Oflazer. Design and Implemetation of of a Spelling Checker for Turkish. *Literary and Linguistic Computing*, 8(3), 1993. Oxford University Press.

[14] Swedish Instute of Computer Science, The Intelligent Systems Laboratory. *SICStus Prolog User's Manual Release 3 Patchlevel 5*, October 1996.

[15] R. Underhill. *Turkish Grammar*. The MIT Press, 4. edition, 1985.

[16] O. Yılmaz. Design and implementation of a verb lexicon and a verb sense disambiguator for Turkish. Master's thesis, Bilkent University, Department of Computer Engineering and Information Science, September 1994.

[17] A. Zaenen and H. Uszkoreit. Overview. In *Survey of the State of the Art in Human Language Technology*, chapter Language Analysis and Understanding. Available at http://www.cse.ogi.edu/CSLU/HLTsurvey/.

Appendix A

The Lexicon Categories

maj	*min*	*sub*	*ssub*	*sssub*
nominal	noun	common		
		proper		
	pronoun	personal		
		demonstrative		
		reflexive		
		indefinite		
		quantification		
		question		
	sentential	act	infinitive	ma
				mak
				yış
		fact	participle	dık
				yacak
adjectival	determiner	article		
		demonstrative		
		quantifier		
	adjective	quantitative	cardinal	
			ordinal	
			fraction	
			distributive	
		qualitative		

Table A.1: The lexicon categories (nominals and adjectivals)

maj	*min*	*sub*	*ssub*	*sssub*
adverbial	direction			
	temporal	point-of-time		
		time-period	fuzzy	
			day-time	
			season	
	manner	qualitative		
		repetition		
	quantitative	approximation		
		comparative		
		superlative		
		excessiveness		
verb	predicative			
	existential			
	attributive			
conjunction	coordinating			
	bracketing			
	sentential			
post-position	nom-subcat			
	acc-subcat			
	dat-subcat			
	abl-subcat			
	gen-subcat			
	ins-subcat			

Table A.2: The lexicon categories (adverbials, verbs, conjunctions, and post-positions)