PROBABILISTIC LEARNING OF TURKISH MORPHOSEMANTICS BY LATENT SYNTAX

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

AHMET ÜSTÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COGNITIVE SCIENCE

SEPTEMBER 2017

Approval of the thesis:

PROBABILISTIC LEARNING OF TURKISH MORPHOSEMANTICS BY LATENT SYNTAX

submitted by **AHMET ÜSTÜN** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Director, **Graduate School of Informatics** _____

Prof. Dr. Cem Bozşahin
Head of Department, **Cognitive Science, METU** _____

Prof. Dr. Cem Bozşahin
Supervisor, **Cognitive Science** _____

Examining Committee Members:

Prof. Dr. Deniz Zeyrek Bozşahin
Cognitive Science Department, METU _____

Prof. Dr. Cem Bozşahin
Cognitive Science Department, METU _____

Assist. Prof. Dr. Cengiz Acartürk
Cognitive Science Department, METU _____

Assist. Prof. Dr. Burcu Can Buğlalılar
Department of Computer Engineering, Hacettepe University _____

Assist. Prof. Dr. Umut Özge
Cognitive Science Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: AHMET ÜSTÜN

Signature :

ABSTRACT

PROBABILISTIC LEARNING OF TURKISH MORPHOSEMANTICS BY LATENT SYNTAX

Üstün, Ahmet

M.S., Department of Cognitive Science

Supervisor : Prof. Dr. Cem Bozşahin

September 2017, 51 pages

The language processing capability of humans is highly dependent on the transparent interface between syntax and semantics which is formalized as the grammar. Morphology also interferes with this interface, in languages having rich morphology such as Turkish. This thesis aims to discover word semantics in Turkish from the compositional morphosemantics by underlying latent syntax. A computational model has been developed to learn a morpheme lexicon in which each morpheme contains semantic information in logical form with a basic syntactic type. A knowledge-free segmentation algorithm based on distributional properties of words is used to extract pseudo-morphemes from words. We utilize a classical probabilistic CCG grammar for lexical learning. Since derivational changes can be handled with lexicalization of words, we employ our model for the inflectional morphemes in Turkish. The model has been tested and results obtained is reported in the thesis with various aspects.

Keywords: morphological parsing, morphosemantics, syntax, CCG

ÖZ

TURKÇE İÇİN MORFOLOJİK ANLAMBİLGİSİNİN GİZLİ SÖZDİZİMİ İLE OLASILIKSAL ÖĞRENİMİ

Üstün, Ahmet
Yüksek Lisans, Bilişsel Bilimler Programı
Tez Yöneticisi : Prof. Dr. Cem Bozşahin

Eylül 2017 , 51 sayfa

Insanların dil işleme yeteneği, dilbilgisi olarak formalize edilen sözdizim ve anlambilim arasındaki arayüze bağımlıdır. Turkçe gibi morfolojisi zengin dillerde, morfoloji bu arayüze müdahale eder. Bu tez morfolojik anlambilgisiden ve kelimelerin içerisindeki örtülü sözdiziminden yola çıkarak ile onların anlamlarını keşfetmeyi amaçlamaktadır. Bu bağlamda, morfemlerin sözdizimsel kategorilerinden ve anlamsal öğelerinden oluşan bir morfem sözlüğü öğrenmek üzere bir model geliştirilmiştir. Kelimelerin içindeki olası morfemleri tespit etmek için kelimelerin dağılımsal özelliklerini kullanan bir bölümlenme algoritması, olası morfemlerin sözlük içindeki ağırlıklarını Öğrenme için ise olasılıksal ulamsal dilbilgisi kullanılmıştır. Yapım ekleri anlamları farklı yeni sözcükler ürettiği için, geliştirilen model çekim ekleri üzerine eğilmektedir. Tez kapsamında model test edilmiş ve sonuçlar farklı yönleri ile rapor edilmiştir.

Anahtar Kelimeler: morfolojik analiz, morfolojik anlambilgisi, sözdizimi, ulamsal dilbilim

To my family

ACKNOWLEDGMENTS

I wish to thank, first and foremost, Prof. Dr. Cem Bozşahin for the encouraging guidance he provided me throughout the entire study. This thesis would not have been possible without his invaluable comments, profound knowledge and friendly conversations.

I am grateful to my co-advisor Assist. Prof. Dr. Burcu Can Buğlalılar for her invaluable guidance, never ending support and patience. I learn the majority of what I know on the computational linguistics from her.

I should express my gratitude to Ferhat Yılmaz Savcı for his understanding and valuable insights regarding academic, professional and social life. He never hesitates to share his knowledge and experience so I feel lucky to work with him.

I would very much to thank to two special friends Murathan Kurfalı and Barış Deniz Sağlam for their support. They became lifelong advisors and companions for me, from the moment I met them.

I wish also want to thank to my dear sister Sedef Üstün for being with me, and supporting me all my life.

I owe my deepest gratitude to my mother and father for their eternal love and courage.

This study is partially funded by TUBITAK (The Scientific and Technological Research Council of Turkey) with grant number 115E464.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Thesis

Native speakers can perfectly comprehend the meaning of phrases or sentences. Whatever they hear or read, they recognize automatically the underlying meaning carried within it. In the course of language acquisition, children are required to learn a correct mapping between linguistic units and corresponding meanings from limited inputs to perform this type of ability. It is known that the smallest meaning bearing units in a natural language are morphemes. Thus, in the language acquisition morphemes are also learned and mapped with their most likely meanings.

The aim of the thesis is to learn morphemes with their correct morphological classes as in language acquisition and to parse the word-forms with their corresponding semantics without performing a morphological analysis. We take, as input, what the child hears and hypothesizes, i.e a form and a meaning; then our model parses them into logical forms to build a morpheme lexicon.

Theory of Universal Grammar (Chomsky, 1975, 1986) proposes that children are innately equipped with a set of mechanisms and constraints called Universal Grammar shared by all human languages, to adapt them to a specific language when they are exposed to the linguistic input. According to Combinatory Categorial Grammar (Steedman, 2000; Steedman & Baldridge, 2011) which is a radically lexicalized theory of natural language grammar, the lexicon is the only resource for language-specific knowledge. Together with a set of combinatory rules and principles that are universal to all languages, a lexicon is enough to project a natural language. Thereby, the lexical learning is the core element in language acquisition.

During language acquisition, the task that children face is to extract the boundaries of linguistic units from speech and to associate them with their lexical representations that have syntactic properties combined with semantics. If we assume that word boundaries are learned earlier by children (Jusczyk, 1999; Thiessen & Saffran, 2003), the only input they have is a pair of sequence of words and contextually available meanings.

Zettlemoyer & Collins (2005) shows how a CCG lexicon with lexical items consisting of a phonological form, a syntactic type and a logical form can be learned from form-meaning pairs of sentences exemplified in 1.1. The study contains a computational model to build a probabilistic CCG lexicon by associating the words with possible categories. Kwiatkowski et al. (2012) also bootstraps learning a CCG lexicon to model language acquisition. These

models treat the words as the smallest linguistic units. However, research conducted in this field shows that children also learn morphemes to understand more complex structure.

(1.1) Surface : `you have another toy`
Meaning : *have'(you', another'(x, toy'(x)))*

In this study, we develop a computational model to learn morphemes along with the semantics, that we call morphosemantics. The system takes pairs of unsegmented words and logical forms as input and builds a weighted CCG lexicon. An example input and expected lexical items are given in 1.2 and 1.3 respectively. Since our model parses words into logical forms, syntactic categories of morphemes become latent variables and the morphology is learned by hidden syntax within the words.

(1.2) Surface : `oyuncaklara`
Meaning : *dative'(plural'(toy'))*

(1.3)
`oyuncak`	:=	N	: toy'
`-lar`	:=	$N\backslash N$	: $\lambda x.plural'(x)$
`-a`	:=	$N\backslash N$	: $\lambda x.dative'(x)$

The problem tackled here is not a simple correspondence problem. That is, it cannot be solved by basic string operations such as matching the morphemes with semantic counterparts following the order they appear. In fact, it is completely a learning problem. The model built within the scope of the current study learns which fragments go with which meanings as well as the order they appear without any information regarding the decompositions of words. Learning the morphosemantics occurs with syntax of CCG since the radical lexicalism in CCG allows language-specific information to be learned along with universal syntax as in language acquisition. The parsing is not constrained by any set of combinators or star modalities, so CCG is used with full power. The model hypothesizes a set of lexical items that has CCG categories assigned with possible substrings i.e. pseudo-morphemes in the words. The search space for hypotheses is narrowed down by a segmentation model which takes the distributional properties of the words into account.

Starting with Chomsky (1970), it is widely acknowledged that derivational morphology is internal to the lexicon. In most cases, its semantics is non-compositional and does not interact with grammatical meanings (Bozsahin, 2002). Thus, the thesis focuses on learning the inflectional morphology. That is, the model is designed to learn the inflections in the words

The lexical learning occurs without any morphological knowledge so the training set does not contain decomposition or any morpheme level clues.[1] Nevertheless only segmented items can take categories in CCG due to transparency of derivation. Therefore, we split the words into pseudo-morphemes by using a segmentation algorithm.

The segmentation algorithm we design is based on the distributional properties of the words that represent the meanings of them. Because the inflectional affixes do not change the meaning of the word radically, a semantically-driven segmentation model is used. The aim of the

[1] The only assumption we accept is that the nominal forms of the words and their syntactic categories are lexicalized before the learning process. See the Chapter 7 for details.

procedure is to generate feasible segmentations of a word and extract possible substrings as pseudo-morphemes. We employ the unsupervised embeddings of words (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Pennington et al., 2014) by using *word2vec* to access distributional features and use them to calculate the corresponding semantic similarity for each character boundary in the segmentation procedure.

We use the classical probabilistic CCG (Zettlemoyer & Collins, 2005) for the statistical learning of morphemic lexicon. The lexicon keeps a weight for each pseudo-morpheme in addition to the surface form, syntactic type and logical form.

1.2 Motivation

The thesis has two main motivations from the cognitive science perspective:

1. To present a model that is designed to learn the words with their morphology as in language acquisition period of children.
2. To demonstrate that morphology can be considered as an internal syntax which forms the compositional meaning in the words.

Since our study aims to provide more realistic setting for the language acquisition and to be more cognitively plausible, the model we develop only uses what children can access from the environment, that is, the compositional and the distributional meaning of the words.

According to the Language of Thought Hypothesis (Fodor, 1975), thought and thinking occurs in a mental language that involves combinatorial syntax with semantics. This makes possible to formalize compositional meaning of words with logical forms. Thus, our model parses these logical forms with the help of combinatorial grammar which is latent in the word structure.

Although our model is designed to be generic, the scope of this thesis is solely Turkish inflectional morphology. Thus, we experimented our model on inflected words in Turkish.

1.3 Outline

The thesis consists of seven main chapters: Morphology, Combinatory Categorial Grammar, Data, Learning Morphology by Latent Syntax, Results, Discussion and Conclusion

Morphology and CCG chapters describe the background of our study. Morphology (Chapter 2) gives detailed information about the history of morphology studies. CCG (Chapter 3) contains a short summary of the CCG required to understand our model. Data and Model chapters explains our model and the data used in the experiments. We discuss the two main arguments of the study in Discussion chapter. The last chapter concludes the thesis.

CHAPTER 2

MORPHOLOGY

2.1 Introduction

In linguistics, morphology is the study of the forms of words, and the study of how words are made up of smaller pieces, i.e., morphemes which are the smallest meaning-bearing elements of a language. Morphology is also studied as an interface between syntax and phonology since the morphological constructions carry information from phonological levels to syntax and semantics. (Katamba & Stonham, 1993)

Morphology entered the domain of generative linguistics with Chomsky (1970), Halle (1973), Siegel (1974), Aronoff (1976). The main issue studied by generative linguists is how morphological representations, which constituted the word forms, interact with representations at other levels such as syntax and semantics. Also, the nature of morphological units and mechanism in the formation of words are the other main questions about morphology. Chomsky (1970) put the morphology as a part of the lexicon. However, when syntactic relations are not included in the account, especially in inflectional morphology, lexical rules of morphology become just the application of phonological changes to stems and affixes. (Cakıcı, 2008)

The initial attempt to describe morphology as a set of morphosyntactic rules was presented by Hockett (1954) in two different approaches; item-and-arrangement hypothesis, item-and-process hypothesis. In item-and-process approach, a word form results from an application of rules which modify the phonological form of the word or the stem. Resulting word forms are not considered as a composition of stem and affixes, operations identified by rules alter the word forms to a sequence of phonemes so they can not be fragmented into smaller part. That is why this approach is called lexeme based morphology.

In item-and-arrangement approach, each word form is analyzed as a set of morphemes arranged in sequence. Example 2.1 shows that Turkish word *arabalar* 'cars' is represented as the combination of root morpheme *araba* 'car' and Turkish plural suffix *lAr* 's' according to item-and-arrangement approach.

$$\begin{aligned} (2.1) \qquad arabalar &\Rightarrow araba + lAr \\ &\Rightarrow root + PLU \end{aligned}$$

Roark & Sproat (2007) stated that both item-and-process and item-and-arrangement approaches are fitting to different languages. For example, the morphology of agglutinating languages

which have a linear sequence of morphemes and systematic morphosyntax such as Turkish and Finnish (see Section 2.1.1 for details) are more appropriate to morpheme based approaches according to them. However, Schmerling (1983) pointed out that the categorial grammar is much more consonant with item-and-process model.

A further approach is "Constraint-based Morphology" (Bird, 1990; Russell, 1993). Generally, constraint-based architecture refers to a grammar which is formed from a set of constraints on possible linguistic objects (Bonami & Crysmann, 2016). The constraint-based approach of morphology uses the phonological properties of the words as a set of constraints that links morphology and syntax which are considered to the separate dimensions of language.

In the sections that follow the above introduction, we concentrate on the structure of Turkish language, the relationship between morphology and syntax and computational models for morphological processing. Since the thesis mainly focuses on Turkish morphology, we will describe the morphology in agglutinating languages before the chapter's main points.

2.1.1 Morphology in Agglutinating Language

According to Spencer (1991) languages are divided into four classes: isolating, inflectional, polysynthetic and agglutinating. Isolating languages are languages with limited or no morphology such as Vietnamese or Chinese. As distinct from isolating ones, in inflectional or fusional languages, morphemes can have multiple grammatical or semantic features that make the language very complex. Greek, Russian and Polish are instances of inflectional languages. Polysynthetic languages have words that consist of many morphemes which can have independent meaning, to form a sentence. Similar to polysynthetic languages, agglutinating languages have relatively rich morphology. Words are formed from different morphemes but each morpheme bears single grammatical feature that affects the meaning of words.

Hankamer (1989) described what characterizes agglutinating languages as follows:

> Stem formation by affixation to previously derived stems is extremely productive, so that a given stem, even though itself quite complex, can generally serve as the basis for even more complex words.

Turkish is an agglutinating language like Finnish or Swahili. The example 2.2 which is taken from (Hankamer, 1989) shows the agglutinations occurring in Turkish morphology. In example 2.2, the affixation starts with causative suffix *dir*, and new word *indir* 'lower' is derived from the root *in* 'descent'. In continuation; passive, ability, negative, second ability, tense and agreement suffixes are attached to the stem in a linear sequence and resulting word *indirilemiyebilecekler* 'they will be able to not be able to be lowered' or 'they will be able to resist being brought down' is obtained.

(2.2)	in	-dir	-il	-e	-mi	-yebil	-ecek	-ler
	descent	-CAUS	-PASS	-ABLE	-NEG	-ABLE	-TENSE	-AGR
	they will be able to not be able to be lowered							

Theoretically, it is possible to produce a word of infinite length since Turkish morphotactics admit for the nested morphological structure. Hankamer (1989) exemplifies that mechanism as follows:

(2.3)	göz	'eye'
	göz -lük	'glasses'
	göz -lük -çü	'seller of glasses'
	göz -lük -çü -lük	'the occupation of oculists'
	göz -lük -çü -lük -çü	'a lobbyist for the oculist profession'
	göz -lük -çü -lük -çü -lük	'the occupation of being a lobbyist for the oculist profession'

The agglutinating character of Turkish is one of the main basis of the background of the thesis since this character forces morphological learning throughout a parsing mechanism. Hankamer (1989) shows that a machinery that uses principles of Turkish morphotactics can produce 1.8 million word forms from one verb root and 9.2 million word forms from one noun root without any recursion. The number of word forms derived from a verb root and a noun root jumps 26.7 million and 216.6 million respectively when one level of recursion is allowed. Considered these numbers with the amount of verb root and noun root in Turkish lexicon, a human can not store whole morphologically complex words in the mind. As a result, we must parse words to learn morphology.

Moreover, the linear sequence in agglutination and segmental structure in Turkish morphology makes it possible to model learning by Combinatory Categorial Grammar (see Chapter 3 for details), because CCG is a linguistic theory that can only assign categories to segmented items due to the transparency of derivation.

2.2 Turkish Morphology

As described in the previsection above, Turkish is an agglutinating language that contains productive affixation capabilities (Oflazer et al., 1994). Affixations occur through inflectional and derivational morphemes. Despite the affixation in Turkish morphology is based on suffixing, there are a small number of unproductive prefixes from foreign origin. According to Oflazer et al. (1994), words containing such prefixes can be lexicalized separately.

Turkish orthography contains 29 characters with 8 vowels (a, e, ı, i, o, ö, u, ü) and 21 consonants (b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y, z) in its alphabet. In order to achieve the vowel harmony, surface forms of words after affixation are determined by morphophonemic rules. Deletion, alternation, and drop rules on the surface of the morphological structure can be operated regarding these rules. Oflazer et al represented phonemes with meta-phonemes for a clear description of the surface form and lexical form of morphemes which is called two-level morphology. Their list of meta-phonemes is given in Table 2.1.

The examples below are taken from Oflazer et al. (1994). In 2.4, the resolving of low-unrounded vowel which is an alternation type morphophonemic operation is occurred on suffix lAr. 2.5 and 2.6 illustrate the consonant changes and vowel drop operations respectively

	Meta-phoneme	Description
1	D	voiced (d) or voiceless (t)
2	A	back (a) or front (e)
3	H	high vowel (, i, u, ü)
4	R	vowel except o, ö
5	C	voiced (c) or voiceless (ç)
6	G	voiced (g) or voiceless (k)

Table 2.1: Phoneme alternations of Turkish and meta-phonemes which are used by Oflazer et al. (1994)

on root morphemes. Full list of the morphophonemic operations in Turkish are described in (Oflazer et al., 1994).

(2.4)	kedi-lAr	cat-PLU
	kedi0ler	kediler
		cats
(2.5)	kitab-cH	book-NtoN(ci)
	kitap0çı	kitapçı
		bookseller
(2.6)	kapa-Hyor	close-PR-CON-3PS
	kapa00ıyor	kapıyor
		he/she is closing

Turkish morphology has a rich inventory in terms of both derivational and inflectional morphemes. In METU-Sabancı Turkish Treebank (Say et al., 2002), there are 106 distinct morphological tags reported. Table 2.2 shows the entire list of morpheme classes. Table 2.3 and 2.4 which are presented in (Oflazer et al., 1994) show the morphemes used in noun inflections and verb inflections respectively[1]. Since we mainly focus on the inflections in Turkish morphology in the thesis, we do not list the derivational morphemes. An extensive report can be found in (Oflazer et al., 1994).

2.3 Morphology and Syntax

Although it is considered that morphology is an interface between phonology and syntax-semantics, the common approach in studies, especially in computational ones, is to separate morphology and syntax as independent processing steps.

[1] We changed the name of the morphological classes in the tables to make them compatible with METU-Sabancı Turkish Treebank

Morphological Tags		
A1pl	Gen	PCIns
A1sg	Hastily	PCNom
A2pl	Imp	PersP
A2sg	InBetween	Pnon
A3pl	Inf	Pos
A3sg	Ins	Postp
Abl	Interj	Pres
Able	JustLike	PresPart
Acc	Loc	Prog1
Acquire	Ly	Prog2
Adj	Narr	Pron
Adv	Neces	Prop
AfterDoingSo	Neg	Punc
Agt	NegP	Ques
Aor	Ness	QuesP
As	Nom	Range
AsIf	NotState	Real
Become	Noun	Recip
ByDoingSo	Num	Reflex
Card	Opt	ReflexP
Caus	Ord	Rel
Cond	P1pl	Related
Conj	P1sg	Since
Cop	P2pl	SinceDoingSo
Dat	P2sg	Stay
DemonsP	P3pl	Time
Desr	P3sg	Verb
Det	Pass	When
Distrib	Past	While
Dup	PastPart	With
Equ	PCAbl	Without
FitFor	PCAcc	WithoutHavingDoneSo
Fut	PCDat	Zero
FutPart	PCGen	

Table 2.2: Morphological tags (classes) in METU-Sabancı Turkish Treebank

Morphemic Representation	Morphological Class	Gloss	Examples
-lAr	PLU	Plural	arabalar, evler
-(H)m	P1SG	1st person singular possessive	arabam, evim
-(H)mHz	P1PL	1st person plural possessive	arabamız, evimiz
-(H)n	P2SG	2nd person singular possessive	araban, evin
-(H)nHz	P2PL	2nd person plural possessive	arabanız, eviniz
-(s)H	P3SG	3rd person singular possessive	arabası, evi
-lArH	P3PL	3rd person plural possessive	arabaları, evleri
-(y)H	ACC	Objective (accusative) case	arabayı, evi
-nH	ACC	Objective case (after 3P poss)	masasını
-(n)Hn	GEN	Genitive case	arabanın, evin
-(y)A	DAT	Dative case	arabaya, eve
-nA	DAT	Dative case (after 3P poss)	masasına
-DA	LOC	Locative case	arabada, evde
-nDA	LOC	Locative case	masasında
-DAn	ABL	Ablative case	arabadan, evden
-nDAn	ABL	Ablative case	masasından
-(y)lA	INS	Instrumental/comitative case	arabayla, evle
-ki	REL	Relative	evdeki, arabadakilerinki

Table 2.3: Inflectional morphemes listed by Oflazer et al. (1994) for Turkish nouns

In fact, research in this domain shows that morphology and syntax should serve as different dimensions of a unified structure Bozsahin (2002); Cakıcı (2008); Sehitoglu & Bozsahin (1996); Bozsahin (2011). In other words, there is an intricate and bilateral association between morphology and syntax. Below examples are taken from Kılıç (2013) to illustrate an impact of morphology to syntax and also semantics. The accusative case marker $-H$ (-ı or $-i$ according to Turkish vowel harmony) can determine the subject and the object of a verb which is important for the sequential meaning as in Example 2.7 and 2.8

(2.7) Köpek adam-ı ısırdı
Dog man-ACC bit
The dog bit the man

(2.8) Köpeğ-i adam ısırdı
Dog-ACC man bit
The man bit the dog

Morphology also has effects on the argument structure. The examples below are from (Cakıcı, 2008) to point the changes in argument structure caused by inflectional morphemes. The causative morpheme generates a transitive structure as in (2.10) from an intransitive verb:

(2.9) Yangın sön-dü

Morphemic Representation	Morphological Class	Gloss	Examples
-(H)n	REFLEX	Reflexive	kapan, kaçın, örtün, vurun, edin
-Hs	RECIP	Reciprocal/Collective	kaçıştır, büzüştür, koşuşmak
-DHr	CAUS	Causative	kaldır, arttır, güldür, sektir
-t	CAUS	Causative	çıkart, küçült
-(H)r	CAUS	Causative	çıkar, batır
-Hl	PASS	Passive	yapılmış, küçüldü
-(H)n	PASS	Passive	vidalandı
-(y)AmA	IMP	Impossible	geleme, kalama
-mA	NEG	Negative	gelme, kalma
-(H)r	AOR	Aorist tense	kalır, bulur, büyür, gelir
-(A)r	AOR	Aorist tense	geçer, kaçar
-(H)yor	PROG	Progressive	geçiyor, kalıyor, buluyor, gülüyor
-DH	PAST	Past tense	kaldı, geçti, buldu, güldü
-mHş	NARR	Narrative past	kalmış, bulmuş, ölmüş
-(y)AcAk	FUTR	Future	kalacak, gelecek, isteyecek
-(y)A	OPT	Optative	gelmiyeydi, kazmıyaydı
-mAlI	NECES	Necessitative	gelmeli, bulmalı, bilmeli
-sA	COND	Conditional	gelse, vursa, bulasa
-yAbil	ABLE	Abilitative	gidebil, kalamayabil
-yAmA	NEG	Negative	abilitative gideme, okuyama
-yAdur	REPEAT	Repeat	gidedur, çalışadur
-yAkal	STAY	Stay	bakakal
-yAyaz	JUSTLIKE	Almost	düşeyaz, unutayaz
-yAgör	SINCE	Ever since	yapagör
-yAgel	SINCE	Ever since	yapagel
-yAkoy	REPEAT	Repeat	alıkoy
-(y)DI	DESR	Past aux	yapsaydı, gelmişti, gelecekti
-(y)mHş	NARR	Dubitative aux	tembelmiş, gitmişmiş, buradaymış
-(y)sA	COND	Conditional aux	buradaysa, bulduysa, gelmişse
-(y)ken	WHILE	Adverbial aux	gelmişken, buradayken
-ArAk	BYDOINGSO	Adverbial aux	bakarak, gelerek
-cAsInA	ASIF	Adverbial aux	bilmişcesine, uçarcasına
-(H)m	A1SG	1st person singular	geldim, bulmuşum
-(H)z	A1PL	Type I 1st person plural	geliriz, bulmuşuz
-k	A1PL	Type II 1st person plural	geldik, baksak
-(sH)n	A2SG	2nd person singular	gelsen, bulursun
-(sH)nHz	A2PL	2nd person plural	gelseniz, bulursunuz
-DHr	COP	Copula	buradadır, gelmişizdir
-z	A3SG	Type II 3rd singular	yapamaz, gelemez

Table 2.4: Inflectional morphemes listed by Oflazer et al. (1994) for Turkish verbs

Fire extinguish-PST-P3SG
The fire extinguished

(2.10) Ben yangın-ı sön-dür-dü-m
I fire-ACC extinguish-CAUS-PST-P1SG
I extinguished the fire

In addition to causativisation, passive forms can also be produced through morphology. The direct object of the verb in (2.11) disappears with the accusative case marker in (2.12 due to the passive morpheme -(H)n.

(2.11) Kahya yüzüğ-ü bul-du
Butler ring-ACC find-PST
The butler found the ring

(2.12) Yüzük bul-un-du
Ring find-PASS-PST
The ring was found

Besides examples above, morphological decomposition of a word may be ambiguous, and the right scope can only be solved by looking at the syntactic structure. The following examples are from Göksel (2006). Possessive markers in Turkish are ambiguous as seen in Example 2.13, 2.14 and 2.15, so they must be identified syntactically.

(2.13) sev-di-k-ler-imiz
like-PST-REL-PLU-P1PL
those who we like/liked

(2.14) sev-en-ler-imiz
like-REL-PLU-P1PL
those who like/liked us

(2.15) köpek sev-en-ler-imiz
dog like-REL-PLU-P1PL
those among us who like/liked dogs

Bozsahin (2002, 2011) states that morphological processing is a part of the syntax and the data in which syntactic processing occurs contains enough evidence for morpheme semantics. We build a model to process morphology by latent syntax in the thesis to integrate morphological processing and syntactic structure based on this perspective. Although the parsing in our model runs at the word level, doing parsing through syntactic principles can be regarded as the first step for point of view above.

2.4 Computational Models for Morphological Analysis

One of the foundations of cognitive science is the idea that the human mind is an information processor. According to this idea, our mind is a computational device that processes the information that comes from the environment and creates internal representations from them to learn, interact or modify the world. We have enough evidence to think that morphology has systematicity which requires computational processing. Especially for the understanding of the languages having rich morphology such as Turkish, morphological processing is inevitable (Hankamer, 1989; Martin & Jurafsky, 2000; Bozsahin, 2002; Cakıcı, 2008; Kılıç, 2013; Cöltekin, 2015). Cakıcı (2008) showed that only for the verb git - *go*, there are 177 instances in the Turkish treebank. Sehitoglu & Bozsahin (1996) described that the system built with a limited number of morphological rules can produce 2800 inflections from 40 Turkish roots. When we think of the whole language, 200 billion distinct entries generated from 20K noun roots and 10K verb roots are needed to list according to Hankamer (1989). Therefore, our lexicon must contain hundreds of billions of entry, if we analyze the language without morphological processing which is untenable.

Morphological processing can be divided into two main tasks: the morphological segmentation and the morphological parsing. The segmentation task denotes splitting words into morphemes and finding morpheme boundaries. However, the parsing operation refers to the full analysis of a word including segmentation, morphological structure, tags of morphemes. In section below, a short listing of research is given for both segmentation and parsing on Turkish morphology.

2.4.1 Morphological Segmentation

As described above, morphological segmentation is the operation of identifying morphemes in the words. Even though there are supervised and unsupervised computational models for the segmentation task, supervised models generally aim to offer full analysis as well as segmentation. Hence, these types of models are revised in Section 2.4.2. In addition to that, since unsupervised learning proposes knowledge free mechanisms, models described in this section are not designed only for Turkish.

One of the first models for the word segmentation was developed by Harris (1955). He made use of the number of successors (*successor variety*) of a letter within a word to obtain the correct split of the word. This method is called the letter successor variety (LSV) model. If successor variety of a letter is high enough, it is likely that the letter is the morpheme boundary in the word according to the algorithm designed by Harris.

Figure 2.1 which is taken from (Can & Manandhar, 2014), illustrates an example of letter trie which is a successor tree. LSV approach was used in further studies to learn morpheme boundaries (Hafer & Weiss, 1974; Déjean, 1998; Goldsmith, 2006; Çöltekin, 2010).

Another well-known system in the domain is *Linguistica* (Goldsmith, 2001, 2006). Linguistica relies on minimum description length (MDL) principles originated from information theory. According to MDL, the best compression of data provides the best representation of data (Rissanen, 1978). According to MDL the best compression of data provides the best representation of data. Linguistica uses a data structure which contains a stem list, an affix list and

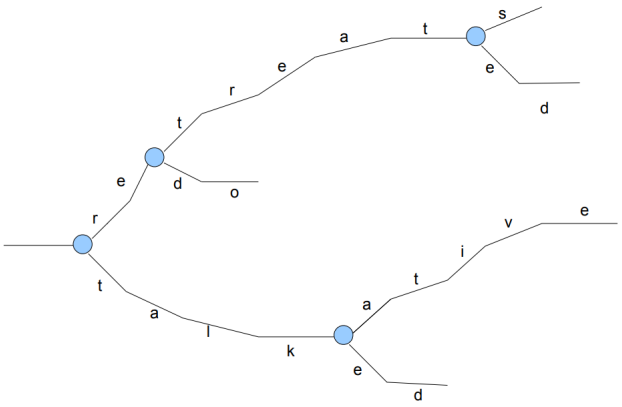

Figure 2.1: Morpheme boundaries in LSV model

a signature list. A signature includes a list of words that have common affixes. Description length (DL) is calculated with respect to the best segmentation model (M) in term of shortness of morphemes, fitting the corpus (C) as follows:

$$DL(C,M) = log_2(\frac{1}{p(C|M)}) \tag{2.1}$$

The probability of a segmentation (w) is based on conditional probability between morphemes and signatures as in Equation 2.3 where t is the root and f is the affix.

$$p(w = t + f) = p(\sigma)p(t|\sigma)p(f|\sigma) \tag{2.2}$$

Morfessor Baseline introduced by Creutz & Lagus (2002) is another model based on MDL principles. The total cost in the model is calculated as follows:

$$\begin{aligned} Cost &= DL(Data) + DL(Codebook) \\ &= \sum_{i \in D} -log\ p(m_i) + \sum_{j \in M} k * l(m_j) \end{aligned} \tag{2.3}$$

where $p(m_i)$ specifies the maximum likelihood estimate of the morpheme m_i, $l(m_j)$ denotes the length of m_j and k represents the number of bits to encode a character.

Besides the baseline model, the Morfessor family also includes Categories ML (Creutz & Lagus, 2004) and Categories MAP (Creutz & Lagus, 2005) models which rely on the maximum likelihood estimation and maximum a posteriori framework respectively. In these models, morphemes are divided into categories that are stem, suffix and prefix. Creutz used the Hidden Markov Models (HMMs) to represent the words with morphemes belonging categories (C) as follows:

$$p(m_1, m_2, ..., m_k) \;=\; [\prod_{i=1}^{k} p(C_i|C_{i-1}) p(m_i|C_i)] \;*\; p(C_{k+1}|C_k) \tag{2.4}$$

where the transition probability between categories is denoted by $p(C_i|C_{i-1})$, the maximum likelihood estimate of morpheme m_i in terms of C_i is calculated by $p(m_i|C_i)$. The last term $p(C_{k+1}|C_k)$ represents the word boundary. Morfessor Categories MAP contains a prior term that encodes the probability of the lexicon in addition the maximum likelihood estimate.

Can & Manandhar (2012) made use of a non-parametric Bayesian model to represent morphological paradigms which are hierarchically clustered. A non-parametric Bayesian model can have infinite-dimensional parameter space to carry information from all possible solutions. Can & Manandhar (2012)'s algorithm recursively operates on hierarchical tree structure to estimate the likelihood of data.

The last state-of-art model reviewed is Morpho Chain proposed by Narasimhan et al. (2015). They used a log-linear model to select correct morphological chains constructed through child-parent relations in the surface form. For example, *hope→hopeful→hopefully* forms a morphological chain, where *hope* is the parent of *hopeful* and *hopeful* is the child of *hope*. The most dominant feature in Morpho Chain is the semantic similarity between words which is based on neural word embeddings.

Unsupervised embeddings of words (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013; Pennington et al., 2014) captures the distributional properties of words in the corpus via neural networks. We used neural word embeddings to find pseudo-morphemes.

2.4.2 Morphological Parsing

In literature, there are various morphological parsers built for Turkish. Most of them are based on a set of rules and finite-state machines (Hankamer, 1986; Oflazer, 1994; Altun & Johnson, 2001; Cöltekin, 2010).

Finite state machines or automata (FSA) are abstract machines having a finite number of states and the conditions that determine the transitions between states through the time. They are frequently used in language modeling (Hopcroft et al., 2001). Figure 2.2 is an example of FSA which is taken from (Kılıç, 2013). Turkish suffixes $-lA$, $-n$ and $-DI$ are represented as transitions in FSA given in 2.2. For example, if we start with Turkish noun sepet *basket*, the verbs sepet-le *to put into basket*" sepet-le-n *to be put into basket* and sepet-le-di, sepet-le-n-di which are forms in the past tense of sepetle and sepetlen can be generated.

A finite state transducer (FST) is an FSA having two tapes. One tape can capture the phonological changes and other tapes can incorporate the morphotactics for morphological modeling as in (Koskenniemi, 1984). Figure 2.3 is presented by Sak et al. (2008) shows a transducer for Turkish vowel harmony that designed according to Turkish phonological rules described in (Oflazer et al., 1994).

Hankamer (1986) combined phonological rules and morphotactics in *keçi* which is the first finite state morphological parser for Turkish. Unlike other FST systems that use the two-level

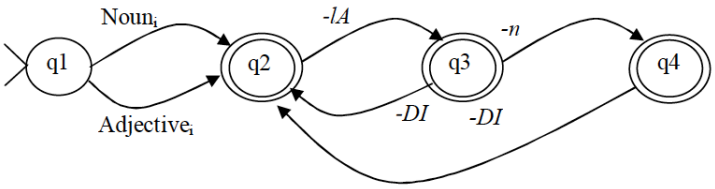

Figure 2.2: A sample FSA for Turkish

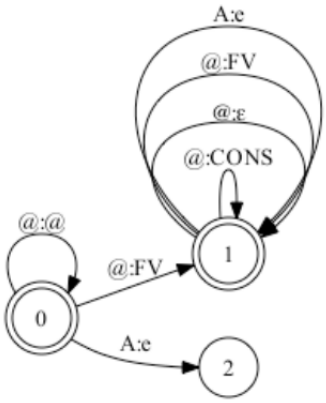

Figure 2.3: An example FST given by (Sak et al., 2008) for Turkish vowel harmony

morphology formalism of Koskenniemi (1984), *keçi* checks the true phonological form of the morpheme at each transition that corresponds a suffixation.

Oflazer (1994) introduced another morphological parser which is implemented according to two-level approach. The parser was designed by using PC-KIMMO (Antworth, 1990) with 23K lexical entry, 22 phonetic rules and Turkish morphotactics described in (Oflazer et al., 1994).

Sak et al. (2008) presented another wide-coverage morphological analyzer for Turkish. They used AT&T FSM tools (Mohri, 1997) with lexicon having 54K root words. A sample output of their parser for the word **alın** is given in Example 2.16. They further performed disambiguation operation by using average perceptron algorithm (Sak et al., 2007) in the same study.

(2.16) alın[Noun]+[A3sg]+[Pnon]+[Nom]
al[Noun]+[A3sg]+Hn[P2sg]+[Nom]
al[Adj]-[Noun]+[A3sg]+Hn[P2sg]+[Nom]
al[Noun]+[A3sg]+[Pnon]+NHn[Gen]
al[Adj]-[Noun]+[A3sg]+[Pnon]+NHn[Gen]
alın[Verb]+[Pos]+[Imp]+[A2sg]
al[Verb]+[Pos]+[Imp]+YHn[A2pl]

CHAPTER 3

COMBINATORY CATEGORIAL GRAMMAR

3.1 Introduction

Combinatory Categorial Grammar (Steedman, 2000; Steedman & Baldridge, 2011; Bozsahin, 2013) is a radically lexicalized linguistic formalism that provides a transparent interface between syntax and underlying semantics where a syntactic parse directly results in an interpretable structure. Steedman & Baldridge (2011) describe CCG with the following definition:

> Combinatory Categorial Grammar (CCG), like other varieties of categorial grammar[...] is a form of lexicalized grammar in which the application of syntactic rules is entirely conditioned on the syntactic type, or category, of their inputs. No rule is structure or derivation dependent.

CCG extends the classical Categorial Grammar (AB) of Ajdukiewicz (1935) and Bar-Hillel (1953). In a categorial grammar, each lexical item is a triplet which contains phonological form, syntactic type and semantic type and it is written as in 3.1. The details are presented in 3.2. Figure 3.3 shows an example of CCG derivation for sentence "*Mary likes musicals*" and the corresponding tree structure for this derivation is given in Figure 3.1.

(3.1) $\texttt{likes} := (S\backslash NP)/NP : \lambda x \lambda y.likes' xy$

(3.2) $\underbrace{\texttt{likes}}_{\textit{surface form}} := \underbrace{\underbrace{(S\backslash NP)/NP}_{\textit{syntactic type}} : \underbrace{\lambda x \lambda y.likes' xy}_{\textit{logical expression}}}_{\textit{category}}$

(3.3)
$$\dfrac{\dfrac{\text{Mary}}{NP : mary'} \quad \dfrac{\dfrac{\text{likes}}{(S\backslash NP)/NP : \lambda x \lambda y.likes' xy} \quad \dfrac{\text{musicals}}{NP : musicals'}}{S\backslash NP : \lambda y.likes' musicals' y}>}{S : likes' musicals' mary'}<$$

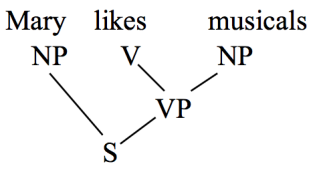

Figure 3.1: The corresponding tree structure for the derivation in the 3.3

3.2 Categories and Combinatory Rules in CCG

3.2.1 Categories

A CCG category can be either atomic or complex. Atomic categories are basic categories having a single item which generally refers to nouns, noun phrases, prepositions or sentences with N, NP, PP, S as in 3.4.

(3.4) $\texttt{Mary} := N : mary'$

Complex categories are made by the composition of atomic categories and other complex categories with slash operators. Slash operators define functions with the parameters on the right of the slash and the result on the left, that is similar to lambda operators in lambda-calculus. For example, 3.5 is a word with syntactic category (NP/N) that expects an N to the right of the word to become an NP.

(3.5) $\texttt{the} := NP/N : \lambda x.x$

In our model, we use N and V for the noun roots and the verb roots. The affixes included in noun inflections and verb inflections are represented with N/N and V/V respectively as CCG categories.

3.2.2 Rules

CCG uses a small set of combinators adopted from combinatory logic to extend the classical categorial grammar because natural languages need more expressive power than the categorial grammar provides. In this respect, we describe composition (**B**), type raising (**T**) in addition to the function application.

Function application is the only rule that the classical categorial grammar has. According to the directionality of application which is determined by slash operators, there are two types of function application rules:

(3.6) Forward Application (>)

$X/Y : f \quad Y : a \;\Rightarrow\; X : fa$

(3.7) Backward Application (<)

$$Y : a \quad X\backslash Y : f \;\Rightarrow\; X : fa$$

Composition which is one of the Curry's combinators (Curry et al., 1958) allows coordination of adjacent strings that can not combine with the function application rules. 3.8 and 3.9 show the classical composition rules. Crossing compositions in 3.10 and 3.11 use to compose crossing dependencies in languages such as Turkish.

(3.8) Forward Composition (>B)

$$X/Y : f \quad Y/Z : g \;\Rightarrow\; X/Z : \lambda x.f(gx)$$

(3.9) Backward Composition (<B)

$$Y\backslash Z : g \quad X\backslash Y : f \;\Rightarrow\; X\backslash Z : \lambda x.f(gx)$$

(3.10) Forward Crossing Composition (>B)

$$X/Y : f \quad Y\backslash Z : g \;\Rightarrow\; X\backslash Z : \lambda x.f(gx)$$

(3.11) Backward Crossing Composition (<B)

$$Y/Z : g \quad X\backslash Y : f \;\Rightarrow\; X/Z : \lambda x.f(gx)$$

Steedman & Baldridge (2011) gives a sample derivation in Figure 3.2 to illustrate the usage of composition rules that yields composite verb *might prove* from *might* and *prove*.

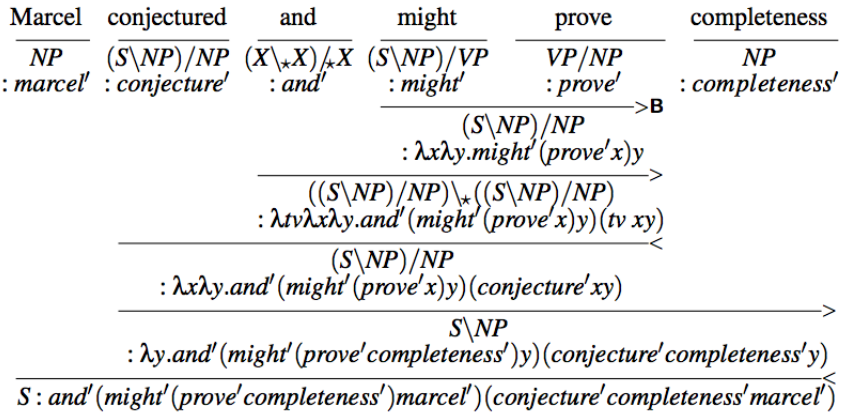

Figure 3.2: A CCG derivation including composition and coordination (Steedman & Baldridge, 2011)

Coordination in the Figure 3.2 is handled by category in 3.12. The $*$ symbol following the slashes allows only the function application on this category.

(3.12) The Conjunction Category

$$\texttt{and} \;:=\; (X\backslash_* X)/_* X$$

Type-Raising transforms arguments to “functions over functions-over-such arguments" by rules given in 3.13 and 3.14. Arguments can compose with verb by using type raising rules. Type-raised arguments expect verbs to join a coordination structure as given in Figure 3.3 (taken from (Steedman & Baldridge, 2011)).

(3.13) Forward Type Raising (>T)

$$X : a \quad \Rightarrow \quad T/(T\backslash X) : \lambda f.fa$$

(3.14) Backward Type Raising (<T)

$$X : a \quad \Rightarrow \quad T\backslash(T/X) : \lambda f.fa$$

In Figure 3.3, *Marcel* and *I* are turned into $S/(S\backslash NP)$ to compose with verbs and become S/NP for coordination by the category of *and*.

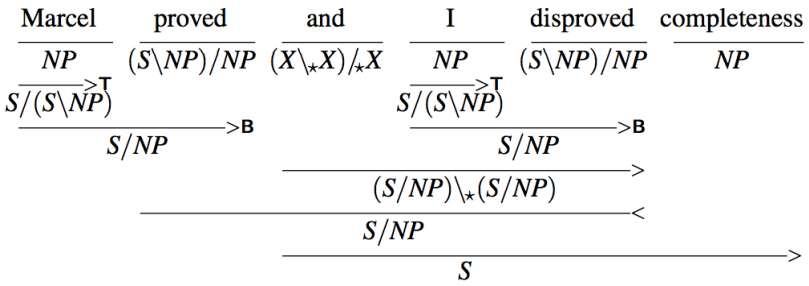

Figure 3.3: A CCG derivation that includes type-raising (Steedman & Baldridge, 2011)

3.3 Morphology and CCG

CCG aims to define a semantically transparent interface on syntactic structures and syntactic derivations. However, when morphology does not take account in the lexicon, some mismatches emerge such as bracketing paradoxes (Bozsahin, 2002). Moreover, from a cognitive science point of view, the ability of the human mind to understand the language can not be explained without morphological processing in languages with rich morphology such as Turkish (Hankamer, 1989; Cakıcı, 2008).

Bozsahin (2002) proposes a morphosyntactic framework based on CCG. He states that morphemes have a phrasal scope rather than word scope, so morphemes affect the semantics of sentences. His morphemic CCG grammar includes the lexical projection of morphosyntactic properties of languages.

Bozsahin (2002) introduces also “morphosyntactic modalities" to manage attachment mechanism in morpheme-based structure. He provides a way to combine phrasal functions of morphemes and rules of morphotactics by these modalities. Figure 3.4 taken from (Bozsahin, 2002) shows an example derivation in morphemic CCG lexicon. The plural marker $-lar$ pluralize the phrase *oyuncak araba*, so resulting parse gives semantically correct construction without distorting the principle of transparency of the derivation.

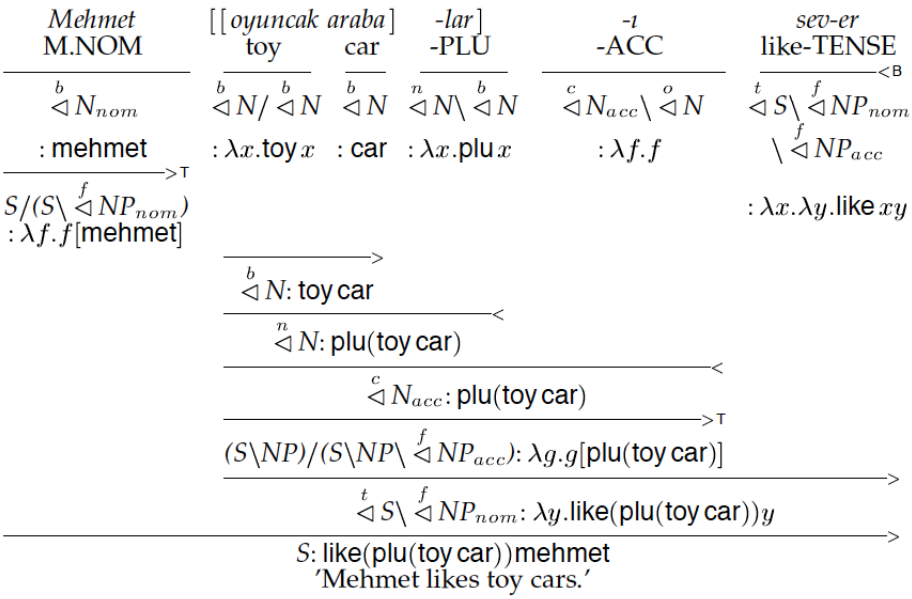

Figure 3.4: A CCG derivation in Morphemic CCG lexicon of Bozsahin (2002)

3.4 Probabilistic CCG (PCCG)

A Probabilistic CCG (Zettlemoyer & Collins, 2005) is a stochastic mechanism to assign probabilities to possible CCG derivations of a sentence and to select the most probable parse by ranking them. Mathematically, it defines a conditional distribution $P(L,T|S)$ over possible (L,T) pairs for a sentence, where L is the logical form and T is the sequence of derivations for s given sentence S.

PCCG uses a conditional log-linear model which is introduced in (Clark & Curran, 2003). For each (L,T,S) triples, a function $\bar{f}$ generates a feature vector in $\mathbb{R}^d$ with d distinct features as in follows:

$$\bar{f}(L,T,S) = \langle f_1(L,T,S), ..., f_d(L,T,S) \rangle \tag{3.1}$$

The formula for the probability of a (L,T) pair for a given sentence S is as follows:

$$P(L,T|S;\bar{\theta}) = \frac{e^{\bar{f}(L,T,S).\bar{\theta}}}{\sum_{(L,T)} e^{\bar{f}(L,T,S).\bar{\theta}}} \tag{3.2}$$

where $\bar{\theta} \in \mathbb{R}^d$ is the parameter vector for a grammar of size n. Therefore, the most probable logical expression (L) for a given sentence S is obtained with the following formula:

$$arg \max_L P(L|S;\bar{\theta}) = arg \max_L \sum_T P(L,T|S;\bar{\theta}) \tag{3.3}$$

The parameter estimation for $\bar{\theta}$ in the training phase with the dataset containing n number of (S_i, L_i) pairs is performed by differentiating the log-likelihood as in 3.4 and 3.5. After the differentiation step, a common stochastic gradient descent algorithm (LeCun et al., 1998) is used to maximize the likelihood.

$$O(\bar{\theta}) = \sum_{i=1}^{n} logP(L_i|S_i;\bar{\theta}) = \sum_{i=1}^{n} log(\sum_{T} P(L_i,T|S_i;\bar{\theta})) \tag{3.4}$$

$$\frac{\partial O_j}{\partial \theta} = \sum_{i=1}^{n}\sum_{T} f_j(L_i,T,S_i)\ P(T|S_i,L_i;\bar{\theta}) - \sum_{i=1}^{n}\sum_{T} f_i(L,T,S_i)\ P(L,T|S_i;\bar{\theta}) \tag{3.5}$$

In our model, we use the CCGlab (Bozsahin, 2017) for probabilistic CCG. The only feature our model has is the lexical feature that represents the number of occurrence of a given lexical entry in a sequence of derivations.

CHAPTER 4

DATA

We used two different datasets: one to learn distributed features of words to be morphologically segmented and other to induced a weighted CCG lexicon to be used in probabilistic morphological parsing. In this chapter, the details of the datasets are presented.

4.1 BOUN Corpus

In order to capture distributed properties of words and to embed them into a low dimensional vector space (see the Chapter 5 for the details), we make use of BOUN corpus provided by Sak et al. (2008). BOUN corpus is a web corpus consists of 423 million word tokens. Different sources of data are used to build this corpus. 184 millions of word tokens are gathered from three newspapers and 239 millions of word tokens are manually crawled from the Turkish web pages. Table 4.1 shows the details of the corpus.

Corpus	Word	Tokens
Milliyet	59M	68M
Ntvmsbnc	75M	86M
Radikal	50M	58M
Web Pages	239M	279M
BOUN Corpus	423M	491M

Table 4.1: Sub-corpora with number of tokens in the BOUN corpus (Sak et al., 2008)

BOUN Corpus contains 48 thousand root words despite a large number of word tokens in it. This is an evidence of productivity in Turkish morphology. A script to remove the punctuation and noisy words is run on the corpus automatically. Corpus is re-organized by removing new lines. The final dataset has 361 million tokens and 725 thousand word types (i.e., distinct word) to be used in our model as given in Table 4.2. This data does not have any label or hint to inform the learning process. The only source of information is the distribution of words.

Final BOUN Corpus	
Word Token	361M
Word Type	275K

Table 4.2: Corpus used for morphological segmentation in our model

4.2 Turkish Corpus of 2016 Sigmorphon Shared Task

The aim of the thesis is to learn the morphemes with their correct morphological classes and to parse the word-forms with corresponding semantics. The semantics of words is determined by the stem and inflectional morphemes attached to it. SIGMORPHON Shared Task (Cotterell et al., 2016) provides the required data to train a probabilistic CCG grammar in our model.

The first SIGMORPHON Shared Task was organized to encourage the computational studies on morphological reinflection in 2016. Systems which joined the shared task tried to generate *reinflections* from inflected words without an explicit morphological analyzer as in the first language acquisition of children. The workshop offered three different tasks regarding the main issue where Task 1 aims to generate inflections, Task 2 and Task 3 demand the reinflection with different restriction and source of information. Datasets from 10 languages including Turkish are provided for each different task.

We use Turkish datasets prepared for the Task 1[1]. The dataset is divided into three portions to be used as training, development (i.e., optimization) and testing data. All portions include the roots of words with their part-of-speech tags. Training and development sets contain a set of morphological classes and inflected forms in pairs. Test set consists of only inflected word forms in addition to word roots. There are no segmented words in these datasets. The full composition of the datasets is given in Table 4.3.

	Pairs	Lemma	Tags
Train	12645	2353	190
Development	1599	1125	170
Test	1598	1128	170

Table 4.3: Turkish datasets provided by SIGMORPHON 2016

In order to ensure that the data used in the lexical training step is entirely contained in the corpus employed for the word embeddings, we remove the words that are not in the BOUN corpus from the training and test sets. We also recreated the sequence of the morpheme classes so that only the classes of the morphemes expressed in the surface form or inflected words remain. The remaining training set consists of 4568 inflected words containing 3808 nouns and 760 verbs. The number of unique word roots in this set is 1353 for the nouns and 260 for the verbs. The test set is obtained by merging the original test set and development set. In

[1] All datasets including the Turkish corpus are publicly available in https://github.com/ryancotterell/sigmorphon2016/tree/master/data/

addition to that, preprocessing operations performed on the training set are also applied on the combined test set and randomly chosen 500 words is used to test. The word with maximum number of suffixes has 4 inflections in both training and test set. The summary of the statistics is given in Table 4.4.

	Training Set			Test Set		
	Noun	Verb	Total	Noun	Verb	Total
# of inflected word forms	3808	760	4568	480	120	500
# of unique root	1353	260	1613	311	90	401
# of words having 4 suffixes	0	10	10	0	4	4
# of words having 3 suffixes	852	170	1022	83	31	114
# of words having 2 suffixes	1805	468	2273	165	83	248
# of words having 1 suffixes	1151	112	1263	132	102	234

Table 4.4: Datasets used for the lexical learning and testing

We make use of CCG for the latent syntax in the internal structure to learn morphology and word semantics. For this reason, each morpheme is defined as a semantic function. Table 4.5 shows the unique functions in the training dataset. The reason why the third person singular agreement marker *A3SG* is not seen in the table is that it is not represented with any morpheme in the surface form of the words. In the further steps of our model (see the Chapter 5), these morpheme types with semantic functions are used the generate CCG entries containing syntactic types as well.

Each training input is a $(surface\ form : semantic\ interpretation)$ pair with the information of the word being a noun or a verb. Input pairs inflected from the Turkish noun acı - *pain* in the training set are given in 4.1.

(4.1)
$$
\begin{aligned}
\texttt{acılarımız} &:= N : possessive1p'(plural'(\mathbf{pain}')) \\
\texttt{acılarda} &:= N : locative'(plural'(\mathbf{pain}')) \\
\texttt{acımızı} &:= N : accusative'(possessive1p'(\mathbf{pain}')) \\
\texttt{acılarımızda} &:= N : locative'(possessive1p'(plural'(\mathbf{pain}'))) \\
\texttt{acılarımızın} &:= N : genitive'(possessive1p'(plural'(\mathbf{pain}'))) \\
\texttt{acıma} &:= N : dative'(\mathbf{pain}') \\
\texttt{acılarınızdan} &:= N : locative'(possessive2p'(plural'(\mathbf{pain}'))) \\
\texttt{acılarınızı} &:= N : accusative'(possessive2p'(plural'(\mathbf{pain}'))) \\
\texttt{acısından} &:= N : ablative'(possessive3s'(\mathbf{pain}')) \\
\texttt{acısına} &:= N : dative'(possessive3s'(\mathbf{pain}')) \\
\texttt{acısında} &:= N : ablative'(possessive2s'(\mathbf{pain}'))
\end{aligned}
$$

PoS	Attribute	Morphological Classes	Semantic Form
Noun	Number	PLU	$\lambda x.plural'(x)$
Noun	Person	PSS1S	$\lambda x.possessive1s'(x)$
Noun	Person	PSS2S	$\lambda x.possessive2s'(x)$
Noun	Person	PSS3S	$\lambda x.possessive3s'(x)$
Noun	Person	PSS1P	$\lambda x.possessive1p'(x)$
Noun	Person	PSS2P	$\lambda x.possessive2p'(x)$
Noun	Person	PSS3P	$\lambda x.possessive3p'(x)$
Noun	Case	ABL	$\lambda x.ablative'(x)$
Noun	Case	ACC	$\lambda x.accusative'(x)$
Noun	Case	DAT	$\lambda x.dative'(x)$
Noun	Case	GEN	$\lambda x.genitive'(x)$
Noun	Case	LOC	$\lambda x.locative'(x)$
Verb	Tense	FUT	$\lambda x.future'(x)$
Verb	Tense	PAST	$\lambda x.past'(x)$
Verb	Aspect	PROG	$\lambda x.progressive'(x)$
Verb	Aspect	PFV	$\lambda x.perfective'(x)$
Verb	Modal	NEG	$\lambda x.negative'(x)$
Verb	Agreement	A1SG	$\lambda x.agreement1s'(x)$
Verb	Agreement	A2SG	$\lambda x.agreement2s'(x)$
Verb	Agreement	A1PL	$\lambda x.agreement1p'(x)$
Verb	Agreement	A2PL	$\lambda x.agreement2p'(x)$
Verb	Agreement	A3PL	$\lambda x.agreement3p'(x)$

Table 4.5: Morphological classes with corresponding semantic form in training data

CHAPTER 5

LEARNING MORPHOLOGY BY LATENT SYNTAX

This chapter describes the computational model for morphology learning. The learning of morphology can be defined as learning of the association between word-forms and their meanings that we call "morphosemantics", since the meaning of a word is composed of the smaller meaning-bearing units, namely morphemes, inside the word. For example, Turkish word arabalarımın - (..) *of my cars* carries both genitive, possessive and plural meaning. Semantics for this word can be represented as follows:

(5.1) $\texttt{arabalarımın} \;:\; genitive'(possessive1s'(plural'(\mathbf{car}')))$

As seen in Example 5.1, we use a logical formalization to represent the semantics of words. Each morphological class acts as a logical function in this formalization (see the Chapter 4 for the full list of morphological classes we employ). An important point is that it is not known which meaning (i.e., morphological class in our case) comes from which substring in the input representation. We expect our model to make correct association between correct part of word and corresponding meaning, with the help of lexicalized grammar and latent syntax.

The logical forms of words that represent their semantics are derived by the syntactic parsing. We make use of CCG to model syntax in the morphology. Each morpheme is defined as a lexical item containing a syntactic type and a logical expression. An example lexicon for the word arabalarımın - (..) *of my cars* and the corresponding CCG derivation are given in 5.2 and 5.3 respectively.

(5.2)
$$\begin{aligned}
&\texttt{araba} &&:= N &&: car'\\
&\texttt{-lar} &&:= N\backslash N &&: \lambda x.plural'(x)\\
&\texttt{-ım} &&:= N\backslash N &&: \lambda x.possessive1s'(x)\\
&\texttt{-ın} &&:= N\backslash N &&: \lambda x.genitive'(x)
\end{aligned}$$

(5.3)
$$\dfrac{\dfrac{\dfrac{\text{araba} \quad \text{-lar}}{N : car' \quad N\backslash N : \lambda x.plural'(x)}<}{N : plural'(car)} \quad \text{-ım}}{N : possessive1s'(plural'(car))} \quad \text{-ın}}{N : genitive'(possessive1s'(plural'(car)))}$$

Morphology is accounted as internal syntax which is the hidden variable in our model to fit a realistic setting for language learning, that is, our model learns morphosemantics of words without any clue as to where the morpheme boundaries are in the word or which part of the word corresponds to which function in the logical form. The model learns the morphological structure of words just from the compositional meaning of them. The underlying assumption for this type of learning is that the one who is exposed to linguistic input through the utterance, has access to the semantics of the input in a logical form as well. The model only accepts that knowledge of word roots that is acquired prior to morphology learning (which is discussed in Chapter 7).

Syntax is the hidden variable in our model and it consists of combinatory categories. Learning of morphology is accomplished during syntactic parsing, which is constrained only by the universal principles of CCG for semantic transparency. All combinators are included in parsing. Combinatory categories of lexical items which are hypothesized in different forms are marginalized over all syntactic derivations. By doing so, the most likely semantic composition of the words is learned by latent syntax.

The assumption that word roots are lexicalized with their syntactic types prior to morphology learning is also dependent on the universal semantics types. Inspired from the Language of Thought (Fodor, 1975), we assume that the learner is able to distinguish the nominal and verbal functions that are composed of semantic type e, $e \rightarrow t$ and $e \rightarrow (e \rightarrow t)$.

To sum up, the target of the model is to learn to map words to logical forms with the definition above. The only input provided to the model is a set of word-meaning pairs illustrated in 5.1. For that reason, our model contains an automatic segmentation algorithm based on neural word embeddings and a lexical learning algorithm using classical probabilistic CCG framework. The workflow of the model is as follow:

1. Model takes a set of $(word : logical\ form)$ pairs.

2. A segmentation algorithm generates the possible segmentations for each word and lists all pseudo-morphemes in the dataset.

3. In order to generate a morpheme lexicon, each pseudo-morpheme is associated with a CCG category regarding to its logical form by template based operation. A limited set of categories is used in this operation. (Steedman & Bozşahin, 2016)

4. For each lexical item a weight is learned though parsing with probabilistic CCG.

Model is tested by comparing the correct logical forms with the derived expressions of words which are not in the training set. Most probable derivation is selected via PCCG for each word which is segmented by the operation in Step 2.

The first section in this chapter briefly describes the segmentation algorithm and the second section covers the lexical learning process which we call $MorphoGenLex$.

5.1 Morphophonological Segmentation by Using Unsupervised Embeddings of Words

Combinatory Categorial Grammar is a linguistic theory that can only assign categories to segmented items, due to the transparency of derivation. The Principle of Combinatory Type-Transparency says that the logical form (semantic type) of a reduction is entirely determined by syntactic types in the derivation (Steedman, 2000). For that reason, words must be divided into smaller substrings before the lexical learning step. Besides that, word segmenting should be performed without any morphological clue to not contradict with the main assumptions in the thesis.

We use a deterministic method based on unsupervised embeddings of words. Word embeddings refer to the vector model learned by neural networks from the distribution of words in a corpus (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013; Pennington et al., 2014). The origins of the idea were introduced by Harris (1954). Harris' distributional hypothesis says that "*linguistic items with similar distributions have similar meanings*". In these types of models, each word is represented by a vector in a low dimensional vector space. It is thought that these vectors learned from the distributions, express the meaning of the words.

Morphological segmentation is one of the natural language processing tasks in which word embeddings are employed (Narasimhan et al., 2015; Soricut & Och, 2015). We used word embeddings to generate possible segmentations of words in the training pairs. The motivation that drives us to use word embeddings is the observation that semantically similar words can be detected by comparing their vectors due to the vectors' capability to represent the contextual meaning of words. Since the inflectional morphemes which we try to learn do not significantly change the meaning of the words, inflected forms of the same stem have higher semantic similarity between them. Hence, an algorithm that uses word embeddings can detect semantic similarity to hypothesize the possible segmentations. The segmentation algorithm proposed in Üstün & Can (2016) is adopted for the current study. Since words are represented by vectors that encode the meanings, semantic similarities between words can be calculated with vectorial distance between them. The form of vectorial distance used in our model is the cosine distance:

$$\cos(v(w_1), v(w_2)) = \frac{v(w_1) \cdot v(w_2)}{||v(w_1)|| \cdot ||v(w_2)||} \tag{5.1}$$

$$= \frac{\sum_{i=1}^{n} v_i(w_1) \cdot v_i(w_2)}{\sqrt{\sum_{i=1}^{n} v_i(w_1)^2} \cdot \sqrt{\sum_{i=1}^{n} v_i(w_2)^2}} \tag{5.2}$$

where $v(w_1)$ and $v(w_2)$ denotes the n dimensional vectors of words w_1 and w_2 respectively

The algorithm takes the input pair $(word : LF)$ and starts with listing all possible segmentations of the given word. The number of segments in a word must be equal to the number of components in the logical form (LF) to provide a semantically transparent derivation[1]. Besides, it is assumed for a given pair that the root of the word is acquired as a lexical information before this segmentation. For the given input pair of word arabalarımın $(..)$ *of my cars*

[1] The underlying assumption is that the resulting reduction in corresponding CCG derivation for a word does not contain a free λ term.

where the root is araba *car*, all possible segmentations are provided in 5.4. The list contains a total of 15 segmentations.

(5.4)

araba-l-a-rımın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-l-ar-ımın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-l-arı-mın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-l-arım-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-l-arımı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-la-r-ımın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-la-rı-mın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-la-rım-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-la-rımı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-lar-ı-mın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-lar-ım-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-lar-ımı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-ları-m-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-ları-mı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-larım-ı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$

After all possible segmentations are listed, our algorithm checks all morpheme boundaries whether the corresponding cosine distance is above a certain threshold. The word (as a character sequence) up to the boundary point and the word up to next boundary point are compared by using the function in 5.1 to calculate the cosine distance for a morpheme boundary. For example, in order to check morpheme boundaries at the segmentation `araba-lar-ım-ın`, the algorithm calculates cosine distance for `(araba,arabalar)` `(arabalar,arabalarım)`, `(arabalarım,arabalarımın)` as shown in Figure 5.1. If a cosine distance for a boundary point is below a threshold, the segmentation that contains this boundary point is removed from the possible segmentation list. The full procedure is presented in Algorithm 1.

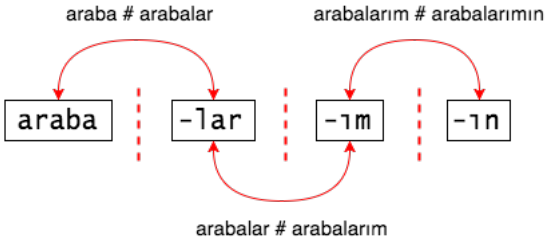

Figure 5.1: The mechanism to check cosine distance at split points for finding morpheme boundaries in `araba-lar-ım-ın`

Üstün & Can (2016) reported that the most feasible cosine distance threshold for Turkish word segmentation task is 0.25. We make use of this value in order to reduce search space for lexical learning. For example, 9 segmentations are eliminated out of 15 segmentations, when the algorithm is performed for the segmentation set given in 5.4. The list in 5.5 shows the remaining segmentations.

Algorithm 1 Automatic segmentation algorithm used to eliminate semantically unfeasible segmentations

procedure AUTOSEGMENT(*word*, *LF* : *word root, threshold*)
    Generate list L for all possible segmentations of *word* w.r.t *LF*
    for all $s \in L$ **do**
        for all b **in** *morpheme boundaries* **do**
            if $cos(v(w_{s,b}), v(w_{s,b+1})) \not> threshold$ **then**
                remove s **from** L
    return L

(5.5)

araba-lar-ı-mın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-lar-ım-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-lar-ımı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-ları-m-ın	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-ları-mı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$
araba-larım-ı-n	:	$genitive'(possessive1s'(plural'(\mathbf{car}')))$

5.2 Inducing a CCG Lexicon by MorphoGenLex

This section describes the learning procedure for morphology in our model. What the model learns is that which substrings (i.e., pseudo-morphemes) are associated with which meanings. The learning procedure we proposed is based on the algorithms used by Zettlemoyer & Collins (2005) and Çoltekin & Bozşahin (2007).

Zettlemoyer & Collins (2005) introduced a lexical learning algorithm GenLex. GenLex takes a sentence and a logical form to generate a set of lexical entries in such a way that at least one parse of the sentence with lexicon is obtained in the results in corresponding logical form. A set of "hand-engineered" rules is defined in GenLex to restrict the number of lexical items generated during the learning phase. Parameters of the lexical items are estimated with PCCG described in Section 3.4.

Çoltekin & Bozşahin (2007) adopted a similar procedure from Zettlemoyer & Collins (2005) to model word acquisition with child directed speech. Their system uses syllable boundaries in the phonetic form of words and concatenates them according to the number of components in the logical form. Syllables (or group of syllables) are associated with basic categories during lexical generation. They employed a Bayesian computation to assign weights to each lexical item obtained.

Unlike the systems above, our model, we call MorphoGenLex, is designed to learn categories of pseudo-morphemes to derive the semantically transparent morphological structure of words, that is, the main focus of our model is syntax in the words. We also do not use manually defined syllable boundaries to segment words. We use distributional properties of words to divide words into pseudo-morphemes and to eliminate a large number of lexical items.

MorphoGenLex begins with an initial CCG lexicon Λ_0 that does not contain any items. As a new input pair arrives, our system first creates a set of segmentations by using the procedure given in Algorithm 1. After the segmenting operation, MorphoGenLex associates a CCG category to each pseudo-morpheme according to logical form in the input pair and a syntactic type selected from the pre-defined templates. The lexical items generated in this way, are placed to the lexicon Λ. We defined two sets of syntactic type templates: one for the nouns and one for the verbs which are given in 5.6 and 5.7 respectively. These templates contain only basic types variated with directionality constraint which is provided by the forward slash ($/$) and the backward operators ($\backslash$). The start modalities ($/_*, \backslash_*$) are not employed not to limit syntactic parsing to only the forward and backward compositions.

(5.6)
$$\begin{aligned} \texttt{syn.type.n1} &:= N \\ \texttt{syn.type.n2} &:= N/N \\ \texttt{syn.type.n3} &:= N\backslash N \end{aligned}$$

(5.7)
$$\begin{aligned} \texttt{syn.type.v1} &:= V \\ \texttt{syn.type.v2} &:= V/V \\ \texttt{syn.type.v3} &:= V\backslash V \end{aligned}$$

MorphoGenLex generates all possible combinations of syntactic types and LF components to create a set of lexical entries for each pseudo-morpheme. The Principle of Combinatory

Type-Transparency guides the process to restrict the number of lexical items for semantic transparency. Since it is assumed that the learner has the knowledge of root word with its syntactic type, corresponding lexical items are automatically generated without any combination. A set of lexical items which are generated from the example word arabalarımın $(..)$ *of my cars* with root araba *car*, is given the Table 5.1 with segmentation list that comes from the previous step.

As in classical PCCG which is described in Section 3.4, each lexical item in lexicon Λ, has a *weight vector* $\bar{\theta} \in \mathbb{R}^d$ in addition to the word form, the syntactic type and the logical form. In this study, we parametrize only the lexical features that keep the number of times the corresponding lexical items is used in the derivation sequence. Therefore each weight is represented by a numerical value[2] In order to estimate the weights, PCCG defines a distribution over parse trees for any word form with the following formula:

$$P(L,T|S;\bar{\theta}) = \frac{e^{\bar{f}(L,T,W).\bar{\theta}}}{\sum_{(L,T)} e^{\bar{f}(L,T,W).\bar{\theta}}} \tag{5.3}$$

where L is the final logical form of the words, T is the sequence of derivations that corresponds to the suffixation in our case and W is the word itself. The most probable logical form (L) for a given word W is obtained with formula in 5.4. In accordance with the problem definition, T represents the hidden syntax in the word structure.

$$arg\ \max_{L} P(L|W;\bar{\theta}) = arg\ \max_{L} \sum_{T} P(L,T|W;\bar{\theta}) \tag{5.4}$$

[2] Initial value of each weight is 1.0 in our model.

1	Input Pair	arabalarımın : $genitive'(possessive1s'(plural'(car')))$ [$gen'(poss1s'(plu'(car')))$]* *These abbreviations are used for morphological classes in the rest of the table	
2	Segmentation Set	araba-lar-ı-mın araba-lar-ım-ın araba-lar-ımı-n araba-ları-m-ın araba-ları-mı-n araba-larım-ı-n	
3	Lexical Set	araba $:= N$: car' -lar $:= N/N$: $\lambda x.plu'(x)$ -lar $:= N/N$: $\lambda x.poss1s'(x)$ -lar $:= N/N$: $\lambda x.gen'(x)$ -ı $:= N/N$: $\lambda x.plu'(x)$ -ı $:= N/N$: $\lambda x.poss1s'(x)$ -ı $:= N/N$: $\lambda x.gen'(x)$ -mın $:= N/N$: $\lambda x.plu'(x)$ -mın $:= N/N$: $\lambda x.poss1s'(x)$ -mın $:= N/N$: $\lambda x.gen'(x)$ -ım $:= N/N$: $\lambda x.plu'(x)$ -ım $:= N/N$: $\lambda x.poss1s'(x)$ -ım $:= N/N$: $\lambda x.gen'(x)$ -ın $:= N/N$: $\lambda x.plu'(x)$ -ın $:= N/N$: $\lambda x.poss1s'(x)$ -ın $:= N\backslash N$: $\lambda x.gen'(x)$ -ımı $:= N/N$: $\lambda x.plu'(x)$ -ımı $:= N/N$: $\lambda x.poss1s'(x)$ -ımı $:= N/N$: $\lambda x.gen'(x)$ -n $:= N/N$: $\lambda x.plu'(x)$ -n $:= N/N$: $\lambda x.poss1s'(x)$ -n $:= N/N$: $\lambda x.gen'(x)$ -ları $:= N/N$: $\lambda x.plu'(x)$ -ları $:= N/N$: $\lambda x.poss1s'(x)$ -ları $:= N/N$: $\lambda x.gen'(x)$ -m $:= N/N$: $\lambda x.plu'(x)$ -m $:= N/N$: $\lambda x.poss1s'(x)$ -m $:= N/N$: $\lambda x.gen'(x)$ -mı $:= N/N$: $\lambda x.plu'(x)$ -mı $:= N/N$: $\lambda x.poss1s'(x)$ -mı $:= N/N$: $\lambda x.gen'(x)$ -larım $:= N/N$: $\lambda x.plu'(x)$ -larım $:= N/N$: $\lambda x.poss1s'(x)$ -larım $:= N/N$: $\lambda x.gen'(x)$	-lar $:= N\backslash N$: $\lambda x.plu'(x)$ -lar $:= N\backslash N$: $\lambda x.poss1s'(x)$ -lar $:= N\backslash N$: $\lambda x.gen'(x)$ -ı $:= N\backslash N$: $\lambda x.plu'(x)$ -ı $:= N\backslash N$: $\lambda x.poss1s'(x)$ -ı $:= N\backslash N$: $\lambda x.gen'(x)$ -mın $:= N\backslash N$: $\lambda x.plu'(x)$ -mın $:= N\backslash N$: $\lambda x.poss1s'(x)$ -mın $:= N\backslash N$: $\lambda x.gen'(x)$ -ım $:= N\backslash N$: $\lambda x.plu'(x)$ -ım $:= N\backslash N$: $\lambda x.poss1s'(x)$ -ım $:= N\backslash N$: $\lambda x.gen'(x)$ -ın $:= N\backslash N$: $\lambda x.plu'(x)$ -ın $:= N\backslash N$: $\lambda x.poss1s'(x)$ -ın $:= N\backslash N$: $\lambda x.gen'(x)$ -ımı $:= N\backslash N$: $\lambda x.plu'(x)$ -ımı $:= N\backslash N$: $\lambda x.poss1s'(x)$ -ımı $:= N\backslash N$: $\lambda x.gen'(x)$ -n $:= N\backslash N$: $\lambda x.plu'(x)$ -n $:= N\backslash N$: $\lambda x.poss1s'(x)$ -n $:= N\backslash N$: $\lambda x.gen'(x)$ -ları $:= N\backslash N$: $\lambda x.plu'(x)$ -ları $:= N\backslash N$: $\lambda x.poss1s'(x)$ -ları $:= N\backslash N$: $\lambda x.gen'(x)$ -m $:= N\backslash N$: $\lambda x.plu'(x)$ -m $:= N\backslash N$: $\lambda x.poss1s'(x)$ -m $:= N\backslash N$: $\lambda x.gen'(x)$ -mı $:= N\backslash N$: $\lambda x.plu'(x)$ -mı $:= N\backslash N$: $\lambda x.poss1s'(x)$ -mı $:= N\backslash N$: $\lambda x.gen'(x)$ -larım $:= N\backslash N$: $\lambda x.plu'(x)$ -larım $:= N\backslash N$: $\lambda x.poss1s'(x)$ -larım $:= N\backslash N$: $\lambda x.gen'(x)$

Table 5.1: All lexical items generated for the word arabalarımın

CHAPTER 6

RESULTS

In this chapter, we describe the results of the experiments that are performed by using the datasets given in Chapter 4. Word embeddings that we make use of in the segmentation algorithm, are obtained by training *word2vec*[1] with a version of BOUN corpus presented in Section 3.1. The lexical learning model is trained with the SIGMORPHON dataset given in Section 3.2. In order to learn a morphemic lexicon with the probabilistic CCG, we use *CCGlab*[2] which is designed by Bozsahin (2017).

The output of our model is a weighted CCG lexicon that contains morpheme entries consisting of a syntactic type, a logical form and a numerical weight. The parser uses this lexicon to derive the most likely logical form for the input word which is segmented into morphemes.

We evaluate our model in terms of two criteria: comprehension of words and coverage of morphemic lexicon. The comprehension task refers that when a word that is not in the training set is given to the system as an input, whether parser grasps the compositional meaning of the word. Moreover, coverage of lexicon shows how much the lexicon covers the actual morphemes in the training set.

6.1 Word Comprehension

In order to evaluate word comprehension success of our model, we give inflected word forms to the system and compare resulting most likely logical forms with the actual logical forms. The design of our model requires segmented word forms to be able to perform semantically transparent derivations (see Chapter 5 for the details of the design). We generate one sequence of segments for each word included in the test set.

The operation of word splitting is based on the algorithm described in Section 5.1. We slightly modify the algorithm so that only one segmentation is obtained for each word. To select one of possible segmentations, we define a normalized distance function $d_n()$ for each of them:

[1] *word2vec* refers to both a neural word embeddings model and also a software program that designed according to model (Mikolov, Sutskever, et al., 2013; Mikolov, Chen, et al., 2013). It was created by a team of researchers led by Tomas Mikolov at Google. We make use of a java implementation to obtain word embeddings (Team, 2017).

[2] *CGGlab* is a software tool implemented in COMMON LISP for experimenting with CCG. We use a model setting with $N(iteration\ numner) = 5$, $\alpha_0 = 0.1$ and $c = 0.1$ as learning parameters

$$d_n(s_i) = \frac{\sum_b cos(v(w_{j-1}),(w_j))}{number\ of\ boundary\ points} \tag{6.1}$$

where s_i is the i^{th} possible segmentation and b denotes the boundary points in s_i. After d_n measure is calculated for each segmentation, the one with the highest d_n is selected as segmented form of the corresponding word.

The dataset used in testing consists of 500 inflected words that include 380 nouns and 120 verbs. After training model with 4568 $(word : LF)$ pairs, our system obtains 364 correct logical forms out of 500. The corresponding accuracy for the comprehension task is 72.80%. Among the 120 verbs in the test set, 101 (84.16%) of them are correctly recognized by our system, whereas only 263 (69.21%) of the nouns out of 380 can be mapped with the correct logical form.

	Noun	Verb	Total
# of Words	380	120	500
# of Correct LFs	263	101	374
Accuracy	69.21%	84.16%	72.80%

Table 6.1: Results of the word comprehension task

When we look at the hit rate of the morphological classes in the logical forms generated by our system according to precision and recall, the model hits 745 morphological classes out of 930 by generating 987 segments. The corresponding values for precision and recall[3] are 75.48% $(745/987)$ and 80.10% $(745/930)$ respectively. The details of the results are given in Table 6.2.

	Noun	Verb	Total
# of Morpheme Classes	694	236	930
# of Predicted Morpheme Classes	535	236	987
# of Correct Morpheme Classes	535	210	745
Precision	73.48%	81.08%	75.48%
Recall	77.08%	88.98%	80.10%
F-Score	74.94%	84.81%	77.42%

Table 6.2: Results for individual hits in logical form predictions

In word comprehension task, our model fails to predict logical forms of 126 words due to three main problems: over-segmentation, under-segmentation and category mismatch. These cases are illustrated in the 6.1, 6.2 and 6.3.

[3] Recall shows the ratio of correct prediction to expected number of predictions while precision refers the correct ones among all prediction made.

(6.1)
$$\dfrac{\dfrac{\dfrac{\text{gel}}{N : come'} \quad \dfrac{\text{-iyo}}{N\backslash N :\ \lambda x.prog'(x)}}{N : prog'(car)}< \quad \dfrac{\text{-r}}{N\backslash N :\ \lambda x.prog'(x)}}{N : prog'(prog'(come))}< \quad \dfrac{\text{-um}}{N\backslash N :\ \lambda x.possessive1s'(x)}}{N : possessive1s'(prog'(prog'(come)))}<$$

(6.2)
$$\dfrac{\dfrac{\dfrac{\text{eldiven}}{N : glove'} \quad \dfrac{\text{-ler}}{N\backslash N :\ \lambda x.plural'(x)}}{N : plural'(glove)}< \quad \dfrac{\text{-ini}}{N\backslash N :\ \lambda x.genitive'(x)}}{N : genitive'(plural'(glove))}<$$

(6.3)
$$\dfrac{\dfrac{\dfrac{\dfrac{\text{göz}}{N : eye'} \quad \dfrac{\text{-ler}}{N\backslash N :\ \lambda x.plural'(x)}}{N : plural'(eye)}< \quad \dfrac{\text{-in}}{N\backslash N :\ \lambda x.possessive2s'(x)}}{N : possessive2s'(plural'(eye))}< \quad \dfrac{\text{-in}}{N\backslash N :\ \lambda x.possessive2s'(x)}}{N : possessive2s'(possessive2s'(plural'(eye)))}<$$

Compared to the syllable-based model developed by (Çoltekin & Bozşahin, 2007) although our model does not use any segmenting information, our model results with similar success for the recognition of words.

6.2 Coverage of The Lexicon

MorphoGenLex which is the lexical generation algorithm in our model aims to map pseudo-morphemes with lexical categories that consist of a syntactic type and a logical form. After it is trained on 4568 input pairs, it generates automatically 5065 lexical entries. 1793 of them are root forms of words and they are provided before the learning. Therefore, MorphoGenLex creates 3272 lexical items from 298 unique substrings.

Each item in the lexicon contains a weight that is initially assigned as 1.0. Lexical training is actually a weight optimization process so that the parser in our model parses words with respect to their logical forms to increase the weights of the correct entries. Table 6.3 shows the top 40 of them according to their weights.

The training set contains 144 unique bound-morphemes. Our model learns all of them with positive weights. However since MorphoGenLex expands the lexicon with all possible substrings, 288 unique lexical items are generated with positive weights for these bound-morphemes. For the verb inflections, the only pseudo-morpheme that gains the dominant weight by mistake is `-r` := $V\backslash V :\ \lambda x.progressive'(x)$. On the other hand, for the noun inflections, (`-ı, -i, -u, -ü`) and (`-ın, -in, -un, -ün`) are confused between *possessive*3*s* - *accusative* and *possessive*2*s* - *genitive* cases respectively due to the phonetic similarity and the lack of morphotactic rules.

Another result of the lexical learning is that our model also learns the allomorphs of the same morpheme jointly in lexicon. For instance, two allomorphs of plural marker, *ler* and *lar*, are lexicalized correctly by the category of $N\backslash N := \lambda x.plural'(x)$ as can be seen in Table 6.3.

Surface Form	Syntactic Type	Logical Form	Weight
-tı	$V\backslash V$	$\lambda x.past'(x)$	15.77
-sı	$N\backslash N$	$\lambda x.possessive3s'(x)$	14.18
-yor	$V\backslash V$	$\lambda x.progressive'(x)$	12.88
-lar	$N\backslash N$	$\lambda x.plural'(x)$	12.79
-dı	$V\backslash V$	$\lambda x.past'(x)$	12.70
-larım	$N\backslash N$	$\lambda x.plural'(x)$	12.51
-nin	$N\backslash N$	$\lambda x.genitive'(x)$	12.48
-acak	$V\backslash V$	$\lambda x.future'(x)$	12.39
-yo	$V\backslash V$	$\lambda x.progressive'(x)$	12.32
-ıyo	$V\backslash V$	$\lambda x.progressive'(x)$	11.98
-ları	$N\backslash N$	$\lambda x.plural'(x)$	11.73
-ıyor	$V\backslash V$	$\lambda x.progressive'(x)$	11.65
-ar	$V\backslash V$	$\lambda x.perfective'(x)$	11.24
-mız	$N\backslash N$	$\lambda x.possessive1p'(x)$	10.95
-la	$N\backslash N$	$\lambda x.plural'(x)$	10.52
-si	$N\backslash N$	$\lambda x.possessive3s'(x)$	10.36
-lar	$V\backslash V$	$\lambda x.agreement3p'(x)$	10.17
-a	$N\backslash N$	$\lambda x.dative'(x)$	10.11
-ımız	$N\backslash N$	$\lambda x.possessive1p'(x)$	10.10
-larımız	$N\backslash N$	$\lambda x.plural'(x)$	9.88
-ler	$N\backslash N$	$\lambda x.plural'(x)$	9.80
-ndan	$N\backslash N$	$\lambda x.ablative'(x)$	9.37
-le	$N\backslash N$	$\lambda x.plural'(x)$	8.96
-dan	$N\backslash N$	$\lambda x.ablative'(x)$	8.94
-lerim	$N\backslash N$	$\lambda x.plural'(x)$	8.75
-e	$N\backslash N$	$\lambda x.dative'(x)$	8.70
-zi	$V\backslash V$	$\lambda x.future'(x)$	8.66
-ım	$N\backslash N$	$\lambda x.possessive1s'(x)$	8.29
-mıyo	$V\backslash V$	$\lambda x.negative'(x)$	8.16
-den	$N\backslash N$	$\lambda x.ablative'(x)$	8.14
-mı	$V\backslash V$	$\lambda x.negative'(x)$	8.08
-larını	$N\backslash N$	$\lambda x.possessive3s'(x)$	7.92
-k	$V\backslash V$	$\lambda x.agreement1p'(x)$	7.75
-üm	$N\backslash N$	$\lambda x.possessive1s'(x)$	7.72
-ma	$V\backslash V$	$\lambda x.past'(x)$	7.71
-rsun	$V\backslash V$	$\lambda x.agreement2s'(x)$	7.71
-da	$N\backslash N$	$\lambda x.locative'(x)$	7.66
-nın	$V\backslash V$	$\lambda x.past'(x)$	7.58
-na	$N\backslash N$	$\lambda x.dative'(x)$	7.42
-du	$V\backslash V$	$\lambda x.past'(x)$	7.29

Table 6.3: The top 40 lexical items in our lexicon

CHAPTER 7

DISCUSSION

The study we present in the thesis contains two main arguments in terms of cognitive science point of view:

1. The model that we design provides a more realistic learning environment in terms of language and specifically word acquisition.

2. The morphological structure of the words has compositional semantics as in the dependency structure of the sentences or phrases. One syntax with combinatorial characteristics can fit for both cases. This should be considered as a linguistic aspect of the Language of Thought Hypothesis (Fodor, 1975) which advocates that "*thought and thinking take place in a mental language*".

The first argument relies on the objective situation in which children are involved during language acquisition. The task they confront is to link phonological input coming from speech to meaning that arises from the environment, that is, they must learn to map part of utterance with constituents of semantics. Kwiatkowski et al. (2012) show that children can learn language from pairs of utterance and logical form representing contextually available meaning. Çoltekin & Bozşahin (2007) make use of a similar procedure to model noun acquisition from child-directed speech. They uses syllabified words to learn to map between syllables (or consecutive clusters of syllables) and morphological knowledge in the noun inflections. We take one step further to show that morphology can be learned by hidden syntax from the distributional and the compositional semantics of words. In our model morpheme boundaries are predicted according to distributional properties of words and morphosemantics is learned by parsing with combinatorial syntax. Distributional properties of words are learned just from raw data in an unsupervised manner.

Results show that distributional properties of words help to map the correct morphemes with the correct semantics. However, our model requires a better algorithm that finds more accurate segmentation of words for recognition of the most likely meaning of words.

In our model, it is assumed that the knowledge of root of words and their syntactic categories (noun or verb) are lexicalized before the learning of the complex structure of words. According to Aksu-Koç (1985), Turkish children capture the nominal case of nouns earlier prior to other morphological variations. Avcu (2014) also showed that children can acquire the knowledge of whether a word is a noun or a verb from various social and attentional cues. Therefore our assumption does not jeopardize the realistic scenario for morphology learning.

The second argument is based on the Language of Thought Hypothesis (Fodor, 1975). LOTH claims that thought and thinking are products of a symbolic system which has combinatorial syntax and semantics, that is, cognitive processes occur in a mental language called *Mentalese* in which tokens are represented with both symbolic representations, that are physically realized in the brain of thinkers. Fodor (1975) suggested that thoughts are the tokens of these representations having a syntactic structure with underlying semantics. Hence, thinking is the process of generating a complex thought from the atomic tokens with the help of syntactic operations defined over these representations. Since thinking governed by combinatorial syntax, resulting products are causally sensitive, i.e., semantically transparent.

According to LOTH, the cognitive ability of language processing has a common foundation with thinking occurs in mentalese. The language is the expression of thought and in order to make assumptions about what thought is being expressed, it should have a compositional structure to form semantics as in mentalese.

Fodor (1975) stated that language learning requires an internal mental language common to all human beings. The language of thought procures ability to make form-meaning association with a combinatorial grammar.

Our study consists of a model to learn morphology by combinatorial syntax, as well. We assume that learner has the ability to make symbolic manipulation on linguistic representations. In our model, morphology is learned from word-meaning pairs by using latent syntax. When considered together with (Zettlemoyer & Collins, 2005) in which syntax is modeled as a hidden variable captured from sentence and logical form pair, our study can be seen as a strong evidence for LOTH.

CHAPTER 8

CONCLUSION

The thesis presents a computational model that learns morphology from the word-form meaning pairs. In the input representation of the model, the meaning of each word is formalized in a logical form without the information about which part of meaning comes from which part of the word. Our model parses morphophonologically segmented words into logical forms by assuming the syntactic categories are latent variables. Thus, morphology is learned by internal syntax which is a hidden variable in our model.

We make use of a small set of CCG categories and principles to internal syntax in the words. The learning occurs via a probabilistic CCG. After the training, our model builds a weighted morphemic CCG lexicon to recognize morpheme and to comprehend words by parsing. Each lexical item in the lexicon contains a surface form (phonological form), syntactic categories, a logical form and a weight. The resulting most likely meaning for a new word is calculated with these weights and possible syntactic derivations. Since CCG is used in the model, it provides a transparent interface between syntax and semantics in lexical projections.

Possible morphophonological segmentation of words is obtained by using distributional properties of words. In order to segment words into pseudo-morphemes with respect to these properties, we train a word embeddings model for words with a neural network, namely *word2vec*. The algorithm we design makes use of cosine similarity between word embeddings to estimate whether a split point can be a morpheme boundary or not.

Therefore, this study aims to learn morphemes with correct morphological classes and to obtain corresponding semantics of any word without morphological analysis. Our input representation and model design are appropriate to language and word acquisition context. In the real scenario, the task that children face is to find linguistic boundaries from phonological streams and to map them with lexical representations. They can only access the phonological and distributional properties of linguistic items and contextually available meanings of them, as in our model. We assume that children sensitive to clues to extract word boundaries from the phonological stream and the root forms of words with basic syntactic type are acquired before the morphological learning.

We perform experiments to test our model on Turkish nominal and verbal inflections. The system is trained with SIGMORPHON dataset described in Chapter 4. The word embeddings model is also trained with BOUN corpus. The results of test are evaluated with respect to two aspects: word comprehension and coverage of the lexicon learned.

Among the 500 words which are not in the training set, our model correctly derive the mean-

ing of 374 words, that is, the accuracy of our model in comprehension task is 72.80%. Although the common belief is that nouns are acquired earlier compared to verbs, our model performs better in verbal inflections. The accuracy of word comprehension in verbs is 84.16% (101/120) whereas nouns have an accuracy of 69.21% (263/380). Avcu (2014) also encountered the same tendency in some of his experiments and reported them in his thesis.

Regarding the results evaluated according to the hit rate of individual morphological classes in logical form predicted by our model, our model finds the correct morphological classes with an f-score of 77.42%, precision of 75.48% (745/987) and recall of 80.10% (745/930). The hit rate of inflections in the verbs is better in this evaluation, as well.

Compared to the Çoltekin & Bozşahin (2007), our system gets competitive results even if the syllables are not provided manually to segment the words. However, ambiguity in the meaning comprehension is higher in our model as expected.

The coverage of lexicon which is learned after the training also gives promising results. Our *MorphoGenLex* algorithm generates 3272 lexical items for the possibly bound-morphemes from 298 unique pseudo-morphemes. 1046 lexical items out of 3272 are increasing in weight as a result of training. The list of real bound-morphemes which consists of 144 morphemes, is entirely contained in 1046 lexical items.

Especially in nominal inflections, there are many phonetically identical bound-morphemes with the same syntactic category combined with different logical form, since we use very limited set of syntactic types to represent them. This causes the ambiguity in comprehension task such as confusion between $possessives3s - accusative$ or $possessive2s - genitive$.

Our model also learns the allomorphs of the same morpheme jointly in lexicon. For instance, two allomorphs of plural marker, *ler* and *lar*, are lexicalized correctly by the category of $N\backslash N := \lambda x.plural'(x)$.

The overall results show that a model that uses both distributional and compositional semantics of words can learn morphology by assuming that there is a latent syntax in word structure.

8.1 Future Work

Firstly, in order to show the universality of the assumption the model can be tested with other concatenative languages such as Finnish or Hungarian.

Secondly, we can increase the number of syntactic types in our lexical template so that our lexicon can also project the morphotactics of the language.

Our model is designed to perform only morphology learning and word comprehension. The last possible extension of our model would be to include the phrase structure to the learning in order to build a fully fledged semantic parser and generate high-coverage morphemic lexicon.

Appendix A

A SAMPLE PARSE RESULT IN CCGLAB

```
Most likely LF for the input: (GEL ECEK LER DI)

 (PST (3PL (FUT GEL))) =
 (PST (3PL (FUT GEL)))

Cumulative weight:  408.94516

Most probable derivation for it: (4 1 403)
------------------------------
LEX     1.0  (GEL) := V
      : GEL
LEX  6.60565  (ECEK) := V\V
      : (LAM X (FUT X))
LEX  4.49914  (LER) := V\V
      : (LAM X (3PL X))
<B   11.1048  (ECEK)(LER) := V\V
      : (LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X)))
LEX  6.59057  (DI) := V\V
      : (LAM X (PST X))
<B   17.6954  (ECEK LER)(DI) := V\V
      : (LAM X
        ((LAM X (PST X)) ((LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X))) X)))
<    66.1909  (GEL)(ECEK LER DI) := V
      : ((LAM X
         ((LAM X (PST X))
          ((LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X))) X)))
        GEL)

Final LF, normal-order evaluated:

   (PST (3PL (FUT GEL))) =
   (PST (3PL (FUT GEL)))

Most weighted derivation : (4 1 403)
------------------------------
LEX     1.0  (GEL) := V
      : GEL
LEX  6.60565  (ECEK) := V\V
      : (LAM X (FUT X))
LEX  4.49914  (LER) := V\V
      : (LAM X (3PL X))
<B   11.1048  (ECEK)(LER) := V\V
      : (LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X)))
LEX  6.59057  (DI) := V\V
      : (LAM X (PST X))
<B   17.6954  (ECEK LER)(DI) := V\V
      : (LAM X
        ((LAM X (PST X)) ((LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X))) X)))
<    66.1909  (GEL)(ECEK LER DI) := V
      : ((LAM X
         ((LAM X (PST X))
          ((LAM X ((LAM X (3PL X)) ((LAM X (FUT X)) X))) X)))
        GEL)

Final LF, normal-order evaluated:

   (PST (3PL (FUT GEL))) =
   (PST (3PL (FUT GEL)))
```

Figure A.1: The parse result for the word “gel-ecek-ler-di”

Bibliography

Ajdukiewicz, K. (1935). Die syntaktische konnexität.

Aksu-Koç, A. A. (1985). The acquisition of turkish. *The Cross-linguistic Studies of Language Acquisition. Vol. 1: The Data*, 839–876.

Altun, Y., & Johnson, M. (2001). Inducing sfa with e-transitions using minimum description length. In *Finite state methods in natural language processing workshop at esslli.*

Antworth, E. L. (1990). Pc-kimmo: a two-level processor for morphological analysis.

Aronoff, M. (1976). Word formation in generative grammar. *Linguistic Inquiry Monographs Cambridge, Mass.*(1), 1–134.

Avcu, E. (2014). Nouns-first, verbs-first and computationally easier first: A preliminary design to test the order of acquisition. *Unpublished master's thesis, Cognitive Science department, Middle East Technical University (ODTÜ), Ankara.*

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language*, *29*(1), 47–58.

Bird, S. (1990). Constraint-based phonology.

Bonami, O., & Crysmann, B. (2016). Morphology in constraint-based lexicalist approaches to grammar. In A. Hippisley & G. Stump (Eds.), *The cambridge handbook of morphology* (p. 588–608). Cambridge University Press. doi: 10.1017/9781139814720.022

Bozsahin, C. (2002). The combinatory morphemic lexicon. *Computational Linguistics*, *28*(2), 145–186.

Bozsahin, C. (2011). Morphological preprocessing or parsing: where does semantics meet computation? In *Foundation of the national speech and language technologies platform workshop on the current status of research on turkish.*

Bozsahin, C. (2013). *Combinatory linguistics*. Walter de Gruyter.

Bozsahin, C. (2017). *Ccglab manual.* `https://bozsahin.github.io/ccglab/CCGlab-manual.pdf`.

Cakici, R. (2005). Automatic induction of a ccg grammar for turkish. In *Proceedings of the acl student research workshop* (pp. 73–78).

Cakıcı, R. (2008). Wide-coverage parsing for turkish. *Doktora Tezi, The University of Edinburgh*.

Çakıcı, R., & Steedman, M. (2009). A wide-coverage morphemic ccg lexicon for turkish. In *Parsing with categorial grammars workshop esslli 2009 bordeaux, france book of abstracts.*

Can, B., & Manandhar, S. (2012). Probabilistic hierarchical clustering of morphological paradigms. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics* (pp. 654–663).

Can, B., & Manandhar, S. (2014). Methods and algorithms for unsupervised learning of morphology. In *Cicling (1)* (pp. 177–205).

Chomsky, N. (1957). Syntactic structures.

Chomsky, N. (1965). Aspects of the theory of syntax.

Chomsky, N. (1970). Remarks on nominalization. *Readings in English Transformational Grammar*, 184–221.

Chomsky, N. (1975). Reflections on language.

Chomsky, N. (1986). *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.

Clark, S., & Curran, J. R. (2003). Log-linear models for wide-coverage ccg parsing. In *Proceedings of the 2003 conference on empirical methods in natural language processing* (pp. 97–104).

Cöltekin, C. (2010). A freely available morphological analyzer for turkish. In *Lrec* (Vol. 2, pp. 19–28).

Çöltekin, Ç. (2010). Improving successor variety for morphological segmentation. *LOT Occasional Series*, *16*, 13–28.

Cöltekin, C. (2015). A grammar-book treebank of turkish. In *Proceedings of the 14th workshop on treebanks and linguistic theories (tlt 14)* (pp. 35–49).

Çoltekin, Ç., & Bozşahin, C. (2007). Syllables, morphemes and bayesian computational models of acquiring a word grammar. In *Proceedings of the cognitive science society* (Vol. 29).

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., & Hulden, M. (2016, August). The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 2016 meeting of sigmorphon.* Berlin, Germany: Association for Computational Linguistics.

Creutz, M., & Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the acl-02 workshop on morphological and phonological learning-volume 6* (pp. 21–30).

Creutz, M., & Lagus, K. (2004). Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th meeting of the acl special interest group in computational phonology: Current themes in computational phonology and morphology* (pp. 43–51).

Creutz, M., & Lagus, K. (2005). Inducing the morphological lexicon of a natural language from unannotated text.

Curry, H. B., Feys, R., Craig, W., Hindley, J. R., & Seldin, J. P. (1958). *Combinatory logic* (Vol. 1). North-Holland Amsterdam.

Déjean, H. (1998). Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning* (pp. 295–298).

Fodor, J. A. (1975). *The language of thought* (Vol. 5). Harvard University Press.

Göksel, A. (2006). Pronominal participles in turkish and lexical integrity. *Lingue e linguaggio*(1), 105–126.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational linguistics*, *27*(2), 153–198.

Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, *12*(4), 353–371.

Goldwater, S., & McClosky, D. (2005). Improving statistical mt through morphological analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 676–683).

Hafer, M. A., & Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information storage and retrieval*, *10*(11-12), 371–385.

Halle, M. (1973). Prolegomena to a theory of word formation. *Linguistic inquiry*, *4*(1), 3–16.

Hankamer, J. (1986). Finite state morphology and left to right phonology. In *Proceedings of the west coast conference on formal linguistics* (Vol. 5, pp. 41–52).

Hankamer, J. (1989). Morphological parsing and the lexicon. In *Lexical representation and process* (pp. 392–408).

Harris, Z. S. (1954). Distributional structure. *Word*, *10*(2-3), 146–162.

Harris, Z. S. (1955). From phoneme to morpheme. *Language*, *31*(2), 190–222.

Hockett, C. F. (1954). Two models of grammatical description. *Word*, *10*(2-3), 210–234.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. *ACM SIGACT News*, *32*(1), 60–65.

Jusczyk, P. W. (1999). How infants begin to extract words from speech. *Trends in cognitive sciences*, *3*(9), 323–328.

Katamba, F., & Stonham, J. (1993). Morphology (modern linguistics series). *London: Tottenham Court Road*, 19–20.

Kılıç, O. (2013). Power of frequencies: n-grams and semi-supervised morphological segmentation in turkish. *Doktora Tezi, Middle East Technical University*.

Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on computational linguistics* (pp. 178–181).

Kwiatkowski, T., Goldwater, S., Zettlemoyer, L., & Steedman, M. (2012). A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics* (pp. 234–244).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Marslen-Wilson, W. (1999). Abstractness and combination: The morphemic lexicon. *Language processing*, 101–119.

Martin, J. H., & Jurafsky, D. (2000). Speech and language processing. *International Edition*, *710*, 25.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational linguistics*, *23*(2), 269–311.

Monson, C., Llitjós, A. F., Ambati, V., Levin, L. S., Lavie, A., Alvarez, A., ... others (2008). Linguistic structure and bilingual informants help induce machine translation of lesser-resourced languages. In *Lrec*.

Narasimhan, K., Barzilay, R., & Jaakkola, T. (2015). An unsupervised method for uncovering morphological chains. *arXiv preprint arXiv:1503.02335*.

Oflazer, K. (1994). Two-level description of turkish morphology. *Literary and linguistic computing*, *9*(2), 137–148.

Oflazer, K., Göçmen, E., & Bozsahin, C. (1994). An outline of turkish morphology. *Report to NATO Science Division SfS III (TU-LANGUAGE), Brussels*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Emnlp* (Vol. 14, pp. 1532–1543).

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465–471.

Roark, B., & Sproat, R. W. (2007). *Computational approaches to morphology and syntax* (Vol. 4). Oxford University Press.

Russell, R. K. (1993). *A constraint-based approach to phonology and morphology* (Unpublished doctoral dissertation). University of Southern California.

Sak, H., Güngör, T., & Saraçlar, M. (2007). Morphological disambiguation of turkish text with perceptron algorithm. *Computational Linguistics and Intelligent Text Processing*, 107–118.

Sak, H., Güngör, T., & Saraçlar, M. (2008). Turkish language resources: Morphological parser, morphological disambiguator and web corpus. In *Advances in natural language processing* (pp. 417–427). Springer.

Say, B., Zeyrek, D., Oflazer, K., & Özge, U. (2002). Development of a corpus and a treebank for present-day written turkish. In *Proceedings of the eleventh international conference of turkish linguistics* (pp. 183–192).

Schmerling, S. F. (1983). Two theories of syntactic categories. *Linguistics and Philosophy*, *6*(3), 393–421.

Sehitoglu, O., & Bozsahin, C. (1996). Morphological productivity in the lexicon. *arXiv preprint cmp-lg/9608015*.

Siegel, D. C. (1974). *Topics in english morphology.* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Siskind, J. M. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, *61*(1), 39–91.

Soricut, R., & Och, F. J. (2015). Unsupervised morphology induction using word embeddings. In *Hlt-naacl* (pp. 1627–1637).

Spencer, A. (1991). *Morphological theory: An introduction to word structure in generative grammar*. Wiley-Blackwell.

Steedman, M. (1996). A very short introduction to ccg.

Steedman, M. (2000). *The syntactic process* (Vol. 24). MIT Press.

Steedman, M., & Baldridge, J. (2011). Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell*.

Steedman, M., & Bozşahin, C. (2016). Projecting from the lexicon.

Steedman, M., & Hockenmaier, J. (2007). The computational problem of natural language acquisition. *Ms., University of Edinburgh*.

Team, D. D. (2017, August). *Deeplearning4j: Open-source distributed deep learning for the jvm, apache software foundation license 2.0.* `http://deeplearning4j.org/`.

Thiessen, E. D., & Saffran, J. R. (2003). When cues collide: use of stress and statistical cues to word boundaries by 7-to 9-month-old infants. *Developmental psychology*, *39*(4), 706.

Üstün, A., & Can, B. (2016). Unsupervised morphological segmentation using neural word embeddings. In *International conference on statistical language and speech processing* (pp. 43–53).

Zettlemoyer, L. S., & Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Conference on uncertainty in artificial intelligence (uai).*

Zettlemoyer, L. S., & Collins, M. (2007). Online learning of relaxed ccg grammars for parsing to logical form. In *Emnlp-conll* (pp. 678–687).