STATISTICAL MORPHOLOGICAL DISAMBIGUATION WITH APPLICATION TO DISAMBIGUATION OF PRONUNCIATIONS IN TURKISH

by
M. OĞUZHAN KÜLEKCİ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Doctorate of Philosophy

Sabancı University
February 2006

STATISTICAL MORPHOLOGICAL DISAMBIGUATION WITH APPLICATION TO DISAMBIGUATION OF PRONUNCIATIONS IN TURKISH

APPROVED BY

Kemal OFLAZER ..... (Thesis Supervisor)

Hakan ERDOĞAN .....

Mehmed ÖZKAN .....

Yücel SAYGIN .....

Berrin YANIKOĞLU .....

DATE OF APPROVAL: .....

STATISTICAL MORPHOLOGICAL DISAMBIGUATION WITH APPLICATION TO DISAMBIGUATION OF PRONUNCIATIONS IN TURKISH

M. Oğuzhan Külekci

EECS, PhD Thesis, 2006

Thesis Supervisor: Prof. Dr. Kemal Oflazer

Keywords: Statistical morphological disambiguation, pronunciation disambiguation, Turkish phrase boundary detection, natural language processing in text-to-speech synthesis

Abstract

The statistical morphological disambiguation of agglutinative languages suffers from data sparseness. In this study, we introduce the notion of *distinguishing tag sets* (DTS) to overcome the problem. The morphological analyses of words are modeled with DTS and the root major part-of-speech tags. The disambiguator based on the introduced representations performs the statistical morphological disambiguation of Turkish with a recall of as high as 95.69 percent. In text-to-speech systems and in developing transcriptions for acoustic speech data, the problem occurs in disambiguating the pronunciation of a token in context, so that the correct pronunciation can be produced or the transcription uses the correct set of phonemes. We apply the morphological disambiguator to this problem of pronunciation disambiguation and achieve 99.54 percent recall with 97.95 percent precision. Most text-to-speech systems perform phrase level accentuation based on content word/function word distinction. This approach seems easy and adequate for some right headed languages such as English but is not suitable for languages such as Turkish. We then use a a heuristic approach to mark up the phrase boundaries based on dependency parsing on a basis of phrase level accentuation for Turkish TTS synthesizers.

BİÇİMBİRİMSEL BELİRSİZLİĞİN İSTATİSTİKSEL GİDERİMİ VE TÜRKÇE OKUNUŞ BELİRSİZLİKLERİNİN ÇÖZÜMÜNDE UYGULANMASI

M. Oğuzhan Külekci

EECS, Doktora Tezi, 2006

Tez Danışmanı: Prof. Dr. Kemal Oflazer

Anahtar Kelimeler: İstatistiksel biçimbirimsel belirsizlik giderimi, Okunuş belirsizliği giderimi, Türkçe sözcük öbeği belirlenmesi, Yazıdan konuşma üretmede kullanılan doğal dil işleme teknikleri

Özet

Eklemeli dillerin biçimbirimsel belirsizliğinin istatistiki olarak giderilmesinde veri yetersizliği problemi belirmektedir. Bu çalışmada bu problemi çözebilmek için *ayırtedici etiket kümeleri* tanımlanmıştır. Kelimelerin biçimbirimsel çözümlemeleri bu kümeler ve kök kelimenin temel etiketi ile modellenmiştir. Geliştirilen sistem Türkçe kelimelerin biçimbirimsel belirsizliğinin istatistiksel olarak giderimini yüzde 95,69'a varan geri çağırım oranlarında başarmaktadır. Yazıdan konuşma üretme sistemlerinde ve akustik ses veri tabanlarının oluşturulmasında kelimelerin olası okunuşları içerisinden doğru okunuşlarının seçilmesi gerekmektedir. Geliştirilmiş olan biçimbirimsel belirsizliği giderici sistem bu problemin çözümüne yönelik olarak kullanılmış, yüzde 99,54 geri çevrim ve yüzde 97,95 kesinlik oranları elde edilmiştir. Yazıdan konuşma üretme sistemlerinde sözcük öbeklerinin belirlenerek vurgunun oluşturulmasında genellikle içerik/görev kelime sınıflandırması kullanılmaktadır. Bu yaklaşım her ne kadar İngilizce ve benzeri diller için uygun olsa da, Türkçe gibi diller için sonuç vermemektedir. Bu nedenle Türkçe metinlerde sözcük öbeklerinin belirlenmesi ve bu öbekler içerisinde de vurgulanacak kelimelerin tesbiti amacı ile de bir buluşsal sunulmaktadır.

Acknowledgements

First, I would like to express my gratitude to my thesis supervisor Kemal Oflazer, not only for his guidance during my study but also for the scientific approach I have learned from him. I would rather be more talented and more hard-working to deserve his supervision, but still I feel myself privileged as his student.

I am greatly indebted to Alparslan Babaoğlu, the Vice President of the National Research Institute of Electronics and Cryptology, for his encouragement and patience. His support of my work was beyond that of a manager.

I am grateful to my thesis committee members Hakan Erdoğan, Mehmed Özkan, Berrin Yanıkoğlu, and Yücel Saygın for their valuable review and comments on the dissertation. Further, Mr.Özkan, who was also my MSc advisor, motivated me to work on natural language processing, and I want to state my appreciation for that support and direction.

I also would like to add that the invaluable kindness and help of Berrin Yanıkoğlu during all five years in Sabancı University, will not be forgotten.

Special thanks to Yasser and İlknur El-Kahlout, Alisher Kholmatov, Özlem Çetin and M. Şamil Sağıroğlu for their friendship and assistance. Their presence has always facilitated my work. In addition, I want to express my thanks to Nancy Karabeyoğlu from Writing Center of the University, for her suggestions to my writing in this dissertation.

Finally, the endless support of my dear wife Şükran has enabled me to finish this study. Words are not enough to indicate even a droplet of her presence in my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NLP	: Natural Language Processing
TTS	: Text to Speech
SAMPA	: Speech Assessment Methods Pronunciation Alphabet
DTS	: Distinguishing Tag Sets
IG	: Inflectional Group
ASR	: Automatic Speech Recognition

Chapter 1

INTRODUCTION

The five main major steps in any natural language processing application along with their basic descriptions are [Covington, 1993]:

- *Phonology* which studies the speech realizations of phonemes in a language and is especially used in text-to-speech synthesis or automatic speech recognition tasks.
- *Morphology* which deals with word analysis and synthesis.
- *Syntax* which deals with sentence structure.
- *Semantics* which deals with meaning in a context.
- *Pragmatics* which integrates the real world knowledge into meaning.

Morphological analysis is an inevitable step of any natural language processing application that requires a serious amount of linguistic analyses such as translation systems, question answering, text understanding, querying in natural language, dialog systems, TTS and ASR systems using text analysis, and so on... Basically, morphological analysis is the task of extracting the inflectional and/or derivational structure of a given word, and assigning tags, which encode the information extracted.

In almost every language, the results of morphological analysis are ambiguous with varying degrees of ambiguity. That is because words can have different analysis of the same orthographic writing. Agglutinative languages with productive word formation and a large number of possible inflections possess high level of ambiguity. Turkish is such a language where approximately 1.8 parses are generated for each word on the average and the tag repository contains over a hundred features to cover its rich morphology. All morphological parses of Turkish word **üstün** are listed below along with their English gloss to demonstrate a general view of the word structure in the language :

1. **üs**`+Noun+A3sg+Pnon+Nom^DB+Verb+Zero+Past+A2sg`, *you were a base*

2. **üstün**+Adj, *superior*

3. **üst**+Noun+A3sg+P2sg+Nom, *your top/clothing/superior*

4. **üst**+Noun+A3sg+Pnon+Gen, *of the top/clothing/superior*

The correct morphological analysis of the word differs depending the context. In sentence **Bu pratikte eşdeğerlerinden üstün bir sistem.** (*This system is superior to its equivalents in practice.*) the second analysis is to be selected, where third one is correct in **Üstün başın paramparça olmuş.** (*Your clothing has been broken into pieces.*). It is essential to select the right morphological analysis in a given context for any further linguistic investigations. Thus, morphological disambiguation is required in many NLP applications.

Morphological disambiguation have been previously studied with statistical, rule-based and hybrid approaches [Brill, 1992], [Oflazer and Tür, 1996], [Ezeiza *et al.*, 1998], [Hajic *et al.*, 2001], [Hakkani-Tür *et al.*, 2002]. Rule based systems are built by writing rules to resolve possible ambiguities. It is difficult to detect all distinct types of ambiguities, and include the related rules. Both the construction and the maintenance of the system tend toward complexity. Rule-based systems generally produce the correct answer if a rule fits the investigated ambiguity. Despite this, they usually fail on situations that have not been encountered before, as no rule has been written to handle them. Thus, in practical applications, where the input is not restricted, they are not preferred. Statisttical systems, on the other hand, are able to handle a wider set of situations, but the accuracy of the disambiguator depends on the language model. The modeling must represent the language well, and its statistical parameters should be extracted from a training set with a high confidence.

The main problem in statistical morphological disambiguation of the languages, which require large feature sets to mark all the morphological properties of words, is *data sparseness*. It is not feasible to find large enough training corpora to extract the whole parameters of a statistical model with a high confidence. Thus, the challenge here is to find a way to represent each morphological analysis by a small number of tags. Prior to this dissertation, Hakkani-Tür *et al.* [2000] proposed to model each syntactic parse of a word by its root word and final inflectional group. The authors reported that they have detected 2194 distinct final IGs in a one million words corpus. They constructed a language model by combining these feature sets with a separate root model.

This dissertation aims to perform the statistical morphological disambiguation of Turkish by using a small number of features and without a need for root language modeling. Distinguishing tag sets are introduced to represent the morphological

parses and a one-million tokens corpus, on which the prior statistical disambiguation work was accomplished, is disambiguated with 374 feature sets without using a root model. We apply the resulting morphological disambiguator to the problem of pronunciation disambiguation. Pronunciation disambiguation refers to the problem of determining the correct pronunciation (phonemes, stress position, etc.) in a given context.

Text-to-speech synthesizers aim to generate the most appropriate speech realization of an input text. Various techniques of natural language processing are used in different steps of a TTS system. Besides the segmentation, tokenization, and text normalization issues, NLP is especially beneficial in generating the correct prosody, essential for high quality natural sounding speech. Morphological analyzers with pronunciation lexicons can be used to perform the grapheme to phoneme conversions appropriately. In addition, the position of the primary stress within a word, which is an important aspect of prosodic structure, can be identified.

It is possible to have more than one phonetic rendering of a word, as each word may have more than one possible reading according to its syntactic or semantic properties in a context. For example the word **karın** has three different pronunciation transcriptions as `/ca:-"r1n/`, `/"ka-r1n/`, and `/ka-"r1n/`.[1] In text-to-speech systems and in developing transcriptions for acoustic speech data, one is faced with the problem of disambiguating the pronunciation of a token in the context used, so that the correct pronunciation can be produced or the transcription uses the correct set of phonemes.

Morphological disambiguation is the main tool for disambiguation of pronunciations. Most of the time it is adequate for detection of the correct pronunciations. However, sometimes the syntactic properties are not enough to differentiate between the readings of a word. For example, the Turkish word **kar** represents such a case. The phonetic transcription should be `/"car/` if it means *profit*, and `/"kar/` if it means *snow*. As the corresponding morphological parses are exactly the same for both meanings, word sense disambiguation must be applied to decide on the pronunciation. Similarly, named entity recognition may be required in some cases, e.g. the primary stress of the word **Gediz** is on the first syllable when it refers a river in Turkey (`/"gj e - d i z/`) and on the second syllable when it is used as a person name (`/gj e - "d i z/`). Thus, besides morphological disambiguation, other techniques are needed for pronunciation disambiguation.

Once individual pronunciations are determined, the phrasal level prosodic context must be considered for more accurate prosody. While reading or talking, the

[1] A detailed investigation of this sample word is given in chapter 3 while explaining the pronunciation disambiguation problem.

speech signals of humans are observed to be divided into phonological phrases separated by longer breaks between some words. To simulate this, TTS systems perform phrase boundary detection on a given text. Although this problem is not totally solved yet, most of the time, heuristics are defined to mark the phrases. These heuristics depend on the grammatical structure of the language. In this dissertation, we also propose heuristics to detect phonological phrases in Turkish. Some words in the detected phonological phrases are to be stressed more than others to correct intonation. An algorithm to perform this intonation is also presented in the suggested phrase detection heuristic.

1.1 Overview

Chapter 2 presents an extensive survey of natural language processing techniques used in text-to-speech synthesis. The first section of the chapter explores the word level NLP issues under the topics of tokenization, vocalization, and morphological analysis. The second section reviews the previous works done on morphological disambiguation and especially the studies performed on Turkish. Pronunciation ambiguities are explained in Section 3 along with the corresponding disambiguation methodologies. The subsections of that section investigate the non-standard word resolution, word sense disambiguation, and named entity recognition tasks. The last section of the chapter is dedicated to advanced linguistic analyses used in phrase level prosodic structures of TTS systems.

Chapter 3 defines the pronunciation disambiguation problem and relates it to morphological disambiguation.

Chapter 4 explores all possible pronunciation ambiguities observed in Turkish language and categorizes the techniques based on them.

Chapter 5 introduces the distinguishing tag sets notion and explains the statistical morphological disambiguation with DTS based modeling.

Chapter 6 details the implementation, disambiguator, and its use in disambiguation of pronunciations in Turkish.

Chapter 7 shows the results of both the morphological and pronunciation disambiguation along with an analysis of the errors.

Chapter 8 proposes the heuristic to find phonological phrase boundaries in Turkish. Besides the detection of the boundaries, an algorithm to identify the stressed words in a phrase is also included.

The thesis ends with the summary and conclusions chapter.

Chapter 2

USE OF NATURAL LANGUAGE PROCESSING IN TEXT-TO-SPEECH SYNTHESIS

Speech synthesis is defined as the realization of an input text in a natural language as speech signals. Such a synthesizer is a full-TTS system if it can automatically convert text written in standard orthography of the language concerned into sound [Shih and Sproat, 1996]. This criteria implies that the input to the system is not in a form of some phonetic transcription but instead a human readable text. An excellent full-TTS synthesizer is expected to read anything that a native speaker of the language can read. In that sense, it is worth noting that the human reader of the text has access to much more information than an automatic speech synthesizer, as he or she certainly *understands* the input. Additionally, the human reader brings experimental and theoretical knowledge at the time of reading and thus has more capabilities to transmit the tone and context of the text to the listeners. Although writing is composed of finite number of graphemes, the speech realizations of those graphemes are infinite [Shalanova and Tucker, 2003]. That is, a written text has infinitely many speech realizations. The text-to-speech (TTS) synthesis problem may be stated as the task of generating the best among those realizations.

The quality of a TTS system is measured using two metrics: *intelligibility* and *naturalness*. The intelligibility of a speech synthesizer is mainly the ability of the system to generate the correct pronunciation for a given word so that the word is understood well when read. The naturalness on the other hand is somewhat a qualitative measurement of the emotion that the TTS system gives to the listener. Improving the naturalness metric be considered as making the system as human-like as possible, so that a listener, say, at the other side of a telephone, will be in doubt as to whether the speech is produced by a TTS system or the reader is a human.

A competition was held in ESCA/COCOSDA'1998,[1] where 17 TTS systems

[1] A workshop of European Speech Communication Association - International Committe for Co-ordination and Standardization of Speech Databases held in Syd-

were evaluated using these metrics. The test results indicate that the intelligibility of nearly all the synthesizers was at an acceptable level with small variations but the naturalness was not that much good [Beutnagel *et al.*, 1999]. Most systems did not perform at an acceptable level on the overall voice quality test perhaps because such systems did not pay the necessary attention to textual and linguistic analyses. For good prosody, a system has to guess which words are to be emphasized and how much [Shih and Sproat, 1996]. A more natural sounding TTS needs more information to be extracted from what is being read. Thus, the improvements need to be performed on the language processing area.

A TTS system has to perform a significant amount of work at *phonological*, *morphological*, *syntactic*, *semantic*, and *pragmatic* levels. Note that those levels are not disjoint and the system has to be seen as a whole. Although most of the recent synthesizers employ NLP on morphological and syntactic levels [Black and Taylor, 1994a, Pfister, 1995, Taylor *et al.*, 1998, Beutnagel *et al.*, 1999, Jilka and Syrdal, 2002, Black and Lenzo, 2003], the same cannot be said for the semantic and pragmatic levels. Problems faced in synthesizer development very much depend on the language. The language independent part of the work is generally in the area of phonetics/acoustics [Shalanova and Tucker, 2003].

The genre of the text on which TTS is deployed is as important as the language of concern [Liberman and Church, 1992, Edgington *et al.*, 1996]. The reading of a dialog is different than the reading of a newspaper. The applications on unrestricted text domains may introduce several problems for a language that the standard orthography of it does not possess. For example, although Turkish does not have a vocalization problem as in Arabic or Hebrew, a synthesizer reading an e-mail, a chat session, or an SMS message in Turkish would most probably need to resolve **slm** as **selam** (*hello*).

Another real life problem confronting TTS systems is mixed-linguality [Pfister and Romsdorfer, 2003]. Most texts in a specific language include foreign words. The inclusion of English words into many world languages or the reading of foreign proper names may cause potential errors in synthesis. Such inclusions are rather frequent and must be handled properly requiring additional resources and processing.

---

ney,Australia. More information about COCOSDA is available at `www.cocosda.org`.

2.1 Why is Natural Language Processing Needed in Text-to-Speech Synthesis?

Before a deeper examination of the natural language processing issues in TTS synthesis, a review of the complete process via some examples may provide a better understanding of the subject.

Although the steps in the synthetic generation of speech are more or less common across languages, some languages introduce problems that are caused by their orthography and writing systems. For those, a certain preprocessing has to be performed. Languages like Chinese which uses no white space require *segmentation*, while languages like Arabic or Hebrew which are written essentially with only consonants, require *vocalization*.

The tokenization problem, which is actually the determination of the syntactic words[2] in a text, is not restricted to languages like Chinese, but many others need it some manner. In English, one needs to resolve **We're** as **We are** or **hasn't** as **has not**, and obviously such occurrences are frequent. However, tokenization in English is very simple when compared to say, Chinese. The Chinese sentence **日文章魚怎麼說** may be tokenized as **日文** (Japanese) **章魚** (octopus) **怎麼** (how) **說** (say) ,or **日** (Japan) **文章** (essay) **魚** (fish) **怎麼** (how) **說** (say) where the first parse *How do you say octopus in Japanese?* is correct [Sproat *et al.*, 1996]. Another language written without word delimiters is Thai and as an example the string ไปหามเหสี in Thai can be segmented into two different ways: ไป (go) หาม (carry) เห (deviate) สี (color) ,or ไป (go) หา (see) มเหสี (queen) [Tesprasit *et al.*, 2003]. It is clear that the meaningful solution is the second one in this case.

The following examples of vocalization have been given by Gal [2002]: The Arabic word كتاب transcribed in Latin characters as **ktb** may correspond to **kitaab** (*books*), or **kuttaab** (*secretaries*). Similarly in Hebrew, **saphar** (*to count*) and **sepher** (*book*), are both written identically with the consonants **spr** ספר.

Apart from the language of concern, the type of application is also an important phenomena while designing a TTS engine, and the style of the input texts may introduce problems that are not present in the standard orthography of the language. An example of this situation is an SMS[3] reader design in Turkish where

[2] Sproat *et al.* [1996] described the orthographic, syntactic and phonetic word discrimination on the example sentence **'I'm going to show up at the ACL'** as: it is composed of eight orthographic words that are separated by seven white spaces, and nine syntactic words with the tokenization of **' I'm '** into **'I am'**, and eleven phonological words if **'ACL'** is to be spelled out while reading.

[3] SMS is the *'Short Messaging Service'* used in mobile phones, which enables the user to send short text messages up to 160 characters long to others.

most people omit the vowels while writing their messages, e.g. they code **bugün ben size gelebilirim** (*I may come to you today*) as **bgn bn sz glblrm**. Although vocalization is not an issue in Turkish, it must be done for a robust SMS message reader in that language.

Morphological analysis is an important issue in TTS synthesis because of two main factors: tagging and word stress assignment. It is especially not feasible to perform part-of-speech tagging without such a component for agglutinative and inflective languages as each word has many different derivations and inflections that can not be compiled into a database. Among the wide range usage of morphological analysis, the following examples are given to express the importance of POS and syntactic tagging: The word **convict** has different pronunciations when used as a verb as in **You convict him** or as a noun in **The convict escaped** [Edgington *et al.*, 1996]. A syntactic analysis (and most probably a context sensitive disambiguation) is to be performed on **They read the book** to resolve whether the verb **read** is in present or past tense.

Text normalization is a crucial step while building a TTS system. In real life the input text to a speech synthesizer often contains non-standard words, such as abbreviations, acronyms, dates, and numbers. These non-standard words have to be converted to a sequence of ordinary words for pronunciation. This mapping includes converting the numbers, dates, e-mail and web addresses, acronyms, abbreviations and various characters such as percentages and currency symbols into words. The discussion of resolving the percent sign (%) in Russian given in the study of Sproat [1997] is a good example on the complexity of this problem. The percent sign maps to different surface forms of word **procent**: with numbers ending in 'one', it is used in nominal form **odin procent** (*one percent*), with numbers 'two', 'three', and 'four', it is rendered in genitive singular form **procenta**; **dva procenta** (*two percent*), or it maps to **procentnaja** in adjectival form **dvadcati-procentnaja skidka** (*twenty-percent discount*), and many other forms exist.

Another example is web address reading in Turkish. Those addresses do not contain the Turkish characters *'ü', 'ö', 'ç', 'ş', 'ı', and 'ğ'*. Thus, the word **hürriyet** is written as **hurriyet** in web address `www.hurriyet.com.tr`, but while reading it is pronounced as in its original form.

Acronyms and abbreviations are also problematic in text normalization process. Some abbreviations introduce ambiguity as they may correspond to more than one word. **Dr.** is such an abbreviation in English, which may either denote *drive* or *doctor*. When pronouncing acronyms, the letters of the acronym may either be spelled out as in **CRC** (*cyclic redundancy check*), or read as a word as in **AIDS**, or maybe all the words referred by the letters are pronounced as in **NY** (New York). Determining how to pronounce them is a serious problem.

Some ordinary words need context sensitive disambiguation for correct reading. As an example the word `row` pronounced differently in **The operations were performed row by row** and in **When the police arrived, the row ended** where it means a *queue* in the former and *fight* in the later. Both interpretations are nouns, and so cannot be resolved by POS tagging. A word-sense disambiguation is required to generate the correct pronunciation.

Generating the pronunciation of proper names (or named entities in other words) differs from ordinary words in two ways: Building a pronunciation lexicon containing all the possible named entities is not possible. Also letter to sound rules may not be consistent with the ones compiled for standard words. Although in practice most frequently seen named entities are collected in a pronunciation lexicon, a robust system has to be able to produce good quality speech for the ones that are not present in that lexicon. Because of these problems the detection of proper names in a text and special processing to generate appropriate pronunciations for them is an important task for TTS systems. The detection process is not a trivial operation even with the information that proper name initials are capitalized in most languages. As an example, the word **apple** refers to the **Apple Computer Inc.** in the sentence **Apple announced a new advance in computer design** versus it is an ordinary noun in **Apple is a nice fruit**. Note also that the type of the named entity may effect the pronunciation in some languages, such as in Turkish, the word **Aydın** has different primary stress assignments in the sentences **Aydın Ege sahillerine yakındır** (*Aydın is near the Aegean coasts.*) and **Aydın zekidir** (*Aydın is intelligent.*), where the word refers to a city in the former and a person's name in the later.

Attention also has to be paid for mixed linguality in a TTS system. The inclusion of foreign words and also foreign proper names occurs very often in most languages. **Swiss Diary** and **World Wide Web** are such inclusions in the example German sentences **Der Konkurs von Swiss Diary** and **Er surft im World Wide Web** [Pfister and Romsdorfer, 2003].

Syntax and semantics greatly influence generating the prosody of a sentence. **I saw the boy in the park with a telescope** may explain different events with different intonations. The observer may have seen the boy who is in the park and carrying a telescope or the observer may have seen the boy, who is in the park, by a telescope or the observer may be sitting in the park and sees the boy with a telescope. A similar example given by Edgington [1996] emphasizes the phrasing with the help of punctuation symbols in the sentence **My husband, who is 27, has left me** versus **My husband who is 27 has left me**. According to the appropriate phrasing, the first includes an explanation regarding the husband, but the second implies that there are more than one husband and the explanation is given

to differentiate between them. Speech synthesizers use the syntactic and semantic constituents in recognition of phonological and intonational phrases both for both a more natural sound and a more accurate rendering of the text.

2.2 Word Level NLP Issues in TTS Synthesis

2.2.1 Preprocessing Tasks

Before the actual conversion of text to speech, some preprocessing on the input text may be required. These tasks can be investigated mainly under the tokenization, vocalization, and non-standard word resolution issues.

Tokenization

The first action to be performed by any application that involves natural language processing is *tokenization* [Webster and Kit, 1992]. Tokenization may be defined as segmentation of an input character string into tokens, which are mainly the words. Guo [1997] gave a very well established formal description of tokenization. Different perspectives on the notions of *word* and *token* from the point of view of lexicography and pragmatic implementations exist. We do not cover those here; see [Webster and Kit, 1992] for discussions. For sake of simplicity, we do not distinguish between word/token and assume that they are interchangeable. As words/tokens are the basic building blocks of a language, further steps of linguistic processing very much depend on this segmentation. The depth and difficulty of a tokenization process depends on two main subjects: the orthography of the language concerned and the application of interest.

Some languages such as Chinese, Japanese and Thai do not have word delimiters. This brings up the importance of the task of tokenization and numerous studies have been published on the subject. An international segmentation contest (The First International Chinese Segmentation Bakeoff) has been held in a recent workshop on Chinese language processing [Sproat and Emerson, 2003].

In most languages, words are delimited by white spaces or punctuation marks. For those, the tokenization task is more related with the type of the application (e.g., a machine translation, information retrieval or a TTS system) rather than the characteristics of the writing system of the language. Different applications may require different standards [Sproat and Emerson, 2003, Sproat *et al.*, 1996]. If one segments the sentence **You've to keep on working** using the space delimiter, then **You've** is just one token, where in reality it is composed of two as **You** and **have**. This information is crucial for a speech synthesis system. From a machine translation perspective, compound words are very important that such a system

needs to mark **"keep on"** as a single entity. Although those are somewhat mixed with morphology and syntax, they are to emphasize all languages somehow need a segmentation process in a varying degree of difficulty according to the writing system and application of concern.

In tokenization there are three main approaches: purely statistical, purely lexical rule-based, and hybrids of the two [Sproat *et al.*, 1996]. Purely statistical methods which rely solely on calculation of probabilities for identifying word boundaries has not gained much interest and it has been indicated that the success of such systems is lower than the purely knowledge-based systems [Webster and Kit, 1992]. Recent works on the topic concentrate on combining knowledge and statistics.

Another point to be decided on is whether the segmenter results a single solution to an input string by using all possible knowledge (morphology, syntax ...) without need for further disambiguation, or the segmenter will detect all possible tokenizations and perform a disambiguation process based on a specified evaluation to choose one of the possibilities [Gou, 1997].

Generally speaking, a tokenization process may be thought of a two phase task. The first phase is the look-up operation from the dictionary where words that form the input string when concatenated one after other, are selected. There may be, and most probably will be, more than one such group of words. If so, the second phase is selecting the right word set from the others (disambiguation).

The ambiguities observed may be *conjunctive* or *disjunctive* [Webster and Kit, 1992]. Let the input string to be segmented as `XYZ` where `X`, `Y`, and `Z` are words from the dictionary. If `XY` is also a word in the dictionary, then the fragment may be segmented as both `X/Y/Z` and `XY/Z` because the compound word `XY` is composed of `X` and `Y`, where each of them is again a word in the dictionary. [4] This is named *conjunctive* ambiguity.

If the input string is as `XsY`, where `X`,`Y`,`Xs`, and `sY` are words in the dictionary and `s` is a string of length bigger than 1, then the second type of ambiguity arises. The fragment may both be tokenized as `Xs/Y` and `X/sY`. That problem is due to the overlapping segment `s`, which may the prefix of word `sY` or the suffix of word `Xs`. This is called *disjunctive* ambiguity.

It is noteworthy here to give the definitions of *critical point* and *critical fragment* [Gou, 1997]. If character c_p in the character array $c_1 \ldots c_p \ldots c_n$ is always a word boundary in all the possible segmentations of the string, then this point is called a *critical point*. The first and last characters are also critical points by definition. The fragment between two critical points is named as *critical fragment*.

[4] Note that slashes ("/") indicate the word boundaries for the examples given in tokenization discussions.

Those critical points are the only unambiguous token boundaries in an input string [Gou, 1997], and the disambiguation process is to be performed on critical fragments. An interesting observation on critical fragments is the *one tokenization per source* has been stated as [Guo, 1998]: "For any critical fragment from a given source, if one of its tokenization is correct in one occurrence, the same tokenization is also correct in all its other occurrences." Informally this observation means that the disambiguations of an ambiguous fragment appearing in different positions in a given context is most probably the same.

The elementary methods in tokenization are modeled by a single framework [Webster and Kit, 1992]. This structural model, called as *Automatic Segmentation Model–ASM(d,a,m)*, classifies those methods by three properties: d for direction of search (right-to-left or left-to-right) for string matching operation, a for addition or omission of characters when words from the dictionary match with some portion of the input string, and m for using principle of minimum or maximum tokenization. To best understand the model, let us examine the *forward* and *backward maximum tokenization* algorithms. The mathematical descriptions of these algorithms and details of the example given below may be found in the related work of Guo [1997].

Let an input string be `ABCD`, and the dictionary `L` be composed of the words `L={ A, B, C, D, AB, BC, CD, ABC, BCD }`. The forward maximum tokenization searches the input string from left-to-right (d parameter of the ASM if left-to-right). The algorithm always tries to match the maximum length dictionary entry always from the left side, which indicates the m parameter of the system is maximum matching.[5] When maximum length word is matched from the left side, the process repeats with the next position until the end is reached. With the forward maximum method, the sample input string `ABCD` is tokenized as `ABC/D`.

The backward maximum tokenization algorithm follows the same procedure of the forward one with the direction reversed. In the backward tokenization algorithm, the matching process is from the right end. The same input is resolved as `A/BCD` with the backward algorithm.

Another well known tokenization is *shortest tokenization*. In the shortest tokenization, the segmentation with the minimum number of words is chosen among the others. For example, the input `ABCBCD` is decomposed as `ABC/BCD` as this segmentation is made up of just two words where the others are of more than two words.

The backward and forward tokenizations can be modeled with ASM very well but the shortest tokenization cannot [Gou, 1997]. Thus, although ASM is a good

[5] For Chinese minimum matching does not work as nearly all characters are stand alone words in the dictionary.

structural model, some methods in the literature exist that do not fit ASM.

It can be stated that each tokenization system somehow performs a look-up operation to match some part of the input string with words in a dictionary. Obviously, the quality of that dictionary impacts on the performance of the whole system [Sproat *et al.*, 1996]. Many words may have different inflections or derivations according to the morphology and storing all forms of all words in a database may be impractical and inefficient. A better way is to construct the tokenization in such a formalism so that the rules of the morphology can be integrated. Moreover, there is a large chance that the system would face out-of-dictionary words such as proper names and foreign words. These unknown words must also be handled, which is best done with statistics. For the disambiguation process, n-gram probabilities of words may be used. The system has to be a mixture of statistical and rule-based approaches to be able to accomplish all these. Sproat *et al.* [1996] propose such a system based on *weighted finite state transducers*. The usage of a finite state methodology is another advantage of the system in that it can be easily integrated to other linguistic parts that are also implemented with finite state techniques, e.g. a finite state morphological analyzer.

Vocalization

In semitic languages, such as Hebrew and Arabic, words are mostly written by only consonants and the vowels are omitted in text. The vowels of a word are defined by the *'pointings'*[6] of its characters indicating missing vowels. Arabic contains 6 such vowel diacritics, and Hebrew 12, although in Hebrew many vowels share the same pronunciation [Gal, 2002]. The example of word **ktb** in Arabic may be vocalized (or pointed) in different ways, such as **kitaab** (*book*), **kutub** (*books*), or **kataba** (*to write*), and many more alternatives also with consonant spreading [Beesley, 1998]. Although the native speakers of those languages do not have serious problems in reading, from the computational linguistics perspective, this ambiguity has to be resolved for any NLP system of those languages [Kamir *et al.*, 2002].

Note that although the semitic languages pose this ambiguity by their nature, there may be similar problems in other languages as well. If we think of an SMS reader in Turkish for example, it is quite highly probable to get an input as **mrhb bgn nslsn?**, which should be converted to **'merhaba bugün nasılsın?'** (*hello*

[6] The vowel characters in Arabic and Hebrew are generally formed by supplying points around the consonant characters. For example the word ערב in Hebrew has the following three versions with different pointings: עֶרֶב, עֲרֵב, עָרְבָ [Kontorovich and Lee, 2001]. Thus, some literature on the subject use the word *'pointing'* referring to vocalization.

how are you today?) in standard orthography of the language for the correct speech synthesis.

The main approaches of that work in Hebrew and Arabic may be investigated in three groups [Kontorovich and Lee, 2001].

The first and the most basic method is just choosing the most frequently seen *diacritisized* (all pointings supplied) form of the input word. It is assumed that the necessary statistics are collected from a fully diacritisized corpus. When the vowels of an unpointed word are to be restored, the most probable pointed version is selected. Note that this method is context-free as it uses just unigram statistics. The success rate for Hebrew is reported as 77% in [Kontorovich and Lee, 2001], 68% in [Gal, 2002], and 74% is given for Arabic in [Gal, 2002] by this baseline method.

The second method is by using the *morphological analysis* information [Choueka and Neeman, 1995, Gal, 2002]. After finding all possible analyses of a word with the corresponding vocalizations, a context-sensitive disambiguation (with some statistics and syntactic rules) is performed among those and the best fitting form is selected. The *Nakdan-Text* [Choueka and Neeman, 1995] system reports a 95% success for Hebrew. The part-of-speech obtained from morphological analysis may also be used in vocalization process. With the knowledge of the POS tags of the input, the most probable diacritisized form of the word with that tag can be selected [Kontorovich and Lee, 2001]. Certainly, this process also needs a training corpus that is fully vocalized and POS-tagged. Kontorovich and Lee [2001] gave the result of such a POS based vocalization as 79% on the Westminister POS-tagged corpus.

The third methodology is by using the *Hidden Markov Models*. As HMMs can include the context sensitive information via the chain of probabilities, they serve as a good basis for vocalization. The unpointed word list is taken as *observations* in the HMM and the hidden states are assigned to pointed forms of those. Given an observation sequence of vowelless words w_1 to w_n, the corresponding hidden state sequence d_1 to d_n holds the vowel-annotated forms of the list. Using this idea as a starting point, Gal [2002] proposed a *bigram* HMM model. This model assumes that a word pointing is dependent on the previous word. The author reported the success rates for the bigram HMM, 81% in Hebrew and 86% in Arabic. The system achieved 87% accuracy in *phonetic group classification* in Hebrew, meaning that; although the right pointings are not restored for a given word, the restored vowels have the same pronunciation with the correct ones. It is noteworthy that the phonetic group classification is a sufficient metric from the TTS point of view.

A slightly different model is tried by Kontorovich and Lee [2001] where there are 14 hidden states. An unpointed observation is emitted one of those 14 states. The reason for using 14 is that the POS based vocalization study reported also in the

same paper used 14 POS tags, and to compare the HMM results with that POS-tag based results, they used that number of states. The parameters of the HMM are trained by Baum-Welch algorithm. The success rate obtained is 81%.

2.2.2 Morphological Analysis

Morphological analysis is the basis of further syntactic and semantic analyses steps in natural language processing. An efficient analyzer is a must, especially for highly inflective and agglutinative languages, where word formations representing different interpretations of a word. In text to speech synthesis such an analyzer may be used in building pronunciation lexicons, in primary stress assignments, in tokenization/normalization processes, and in part-of-speech tagging. Note that with its usage in POS tagging, the morphological analysis component serves as a basic building block for phrase boundary detections, which is essential for generating prosody in a sentence.

Pronunciation lexicons are the lexicons used to convert the graphemic representation of a given word into its pronunciation representation in some form [Oflazer and Inkelas, 2003], so that the input word can be realized as a speech signal. The corpus based word lists for use in speech applications of agglutinative languages, such as Turkish, are inadequate as the rich derivational capability and high number of inflections resulting essentially infinite lexicon [Oflazer and Inkelas, 2003]. It is thus a good idea to encapsulate morphological analysis for pronunciation generation. By using finite state techniques, an input Turkish word is decomposed into all possible morphological analyses and the corresponding pronunciations of each are encoded in SAMPA standard by Oflazer and Inkelas [2003] . As an example, the input word **karın** is resolved into following parses, each containing both the tags between '+' signs as a result of the morphological analysis and also the pronunciation representations given between '(' , ')' brackets:

1. *(ca:-"r)* kar+Noun+A3sg*(1n)*+P2sg+Nom, { *your profit* }
2. *(ka-"r)* kar+Noun+A3sg*(1n)*+P2sg+Nom, { *your snow* }
3. *(ca:-"r)* kar+Noun+A3sg+Pnon*(1n)*+Gen, { *of the profit* }
4. *(ka-"r)* kar+Noun+A3sg+Pnon*(1n)*+Gen, { *of the snow* }
5. *("ka-r)* kar+Verb+Pos*(1n)*+Imp+A2sg, { *mix it!* }
6. *(ka-"r1)* karı+Noun+A3sg*(n)*+P2sg+Nom, { *your wife* }
7. *(ka-"r1n)* karın+Noun+A3sg+Pnon+Nom, { *belly* }

A TTS engine in Turkish has to select one of those above to read the word **karın**. This action obviously requires a disambiguation process. As morphological decomposition tags are also included in the results, a morphological disambiguation can be performed to choose the best for a given context. Note that, although there are seven different morphological parses of the word, there are only three different pronunciations as `ca:-"r1n`, `ka-"r1n`, and `"ka-r1n` corresponding to analyses (1,3), (2,4,6.7), and 5 respectively. Thus, for a speech synthesizer, contrary to the morphological disambiguation process between 7 items, the selection is among just three. Generally, the disambiguation for speech synthesis is a little bit easier than a full morphological disambiguation.

A complete linguistic analysis for an Italian text-to-speech system [Ferri *et al.*, 1997] is a good example of the morphology usage for word pronunciation generation. The system has a three phase structure: morphological analysis, phonetic transcription, and morpho-syntactic parsing. The morphological analyzer obtains the morphological and syntactic features of each word in a given text. Based on this information, the phonetic transcription level marks the stressed syllables and performs grapheme to phoneme conversion. The syntactically related words are grouped so as to generate intonations of prosody at the last morpho-syntactic analysis level. Morphology lies at the heart of the system as both phonetic and syntactic phases depend on the morphological analysis of the first stage.

Despite the somewhat mandatory usage of morphological decomposition in agglutinative languages, other languages, such as English, may also benefit from morphology. An English pronunciation lexicon for speech synthesis was designed by Fitt [2001] which includes morphological breakdowns in the lexicon and addresses the advantages of that morphological annotation. As an historical remark, the *MITalk* English synthesizer [Allen *et al.*, 1987] involved a module, *Decomp*, for morphological decomposition of running text. The *Decomp* model parses an input word into morphemes of three types as prefixes, roots, and suffixes. Each type has also subcategories; the suffixes and prefixes were classified into three levels, and the morphological decomposition by using those levels aimed to assign the correct primary stress for the given word [Church, 1986]. More detailed explanations on that historical approach for stress assignment using morphological analyses can be found in studies of Church [1985, 1986].

Morphological analysis components may also be used on text normalization processes such as homograph resolutions and in POS taggings. The usages will be reviewed in the next sections where related. Note that it may also be used in word segmentation algorithms. A Turkish word segmentation technique using a morphological analyzer was given by Külekci and Özkan [2001], which takes a sequence of words concatenated one after other without word delimiters as input,

and detects all the possible segmentations by using a morphological analyzer.

2.3 Morphological Disambiguation

Each word in a text involves morphologic, syntactic, semantic and even pragmatic information that has to be interpreted somehow for natural language processing. Those information kept in words may be categorized into several classes where each class is represented by a *tag*. Thus, *tagging* may be seen as a kind of classification performed by labeling the information in words with a designated set of tags. Every tag encodes a specific information and a word may take many tags as it may have various distinct properties (part-of-speech, singularity/plurality, tense, personality, case, possessiveness information and etc ...) .

In almost every language, tagging causes ambiguities that have to be resolved by *disambiguation* processes. Stochastic and knowledge-based approaches have been deployed on the problem in the last decade [Brill, 1992], [Oflazer and Tür, 1996], [Hakkani-Tür *et al.*, 2002] and the best results were obtained in hybrid systems that benefit from both [Ezeiza *et al.*, 1998], [Hajic *et al.*, 2001]. Tapanainen and Voutilainen [1994] gives a good discussion about combining the statistical and rule-based systems and Hajic and Hladka [1997] compares the two approaches deployed on Czech language.

Knowledge-based approach requires some rules that will be used to judge on ambiguities. Those rules may be hand-crafted [Oflazer and Kuruöz, 1994], but they are hard to build and difficult to maintain. They can also be learned from a corpus in a supervised [Daelemans *et al.*, 1996] or unsupervised [Oflazer and Tür, 1996] (,which actually combines both hand-crafted rules and unsupervised learning) manner. Brill [1995] proposed a different way of supervised rule extraction that is named as transformation-based error-driven learning and applied to POS tagging of English. Constraint-grammar formalism was introduced by Karlsonn [1990] and has been used in many studies [Voutilainen and Tapanainen, 1993], [Oflazer and Kuruöz, 1994], [Ezeiza *et al.*, 1998], [Tür *et al.*, 1998]. The advantage of rule-based disambiguation is that the choices made by rules between different parses of words are nearly correct all the time. On the other hand, it is difficult to build up such a disambiguator that contains rules for every situation. Hajic *et al.* [2001] argued that performing full disambiguation of *unrestricted* –difficult– texts by rules is a hard task as rule writing and maintaining requires deep linguistic expertise and knowledge-based approaches are good for eliminating incorrect analyses rather than deciding on the best. Another difficulty is the conflicting rule ordering requirements and constraints, thus the sequence of the rules to be deployed on a context effects the result. Oflazer and Tür [1997], and Tür and Oflazer [1998] proposed a so-

lution based on *voting* constraints for the problem. Generally, rule-based approaches do not guarantee to come up with a decision on every type of ambiguities, but if a decision is concluded it is most of the time correct [Tapanainen and Voutilainen, 1994].

Stochastic studies on morphological disambiguation focus on using n-gram language models in HMMs [Hakkani-Tür *et al.*, 2002]. Ratnaparkhi [1996] has introduced a maximum entropy model and Heemskerk [1993] has used *probabilistic* context-free grammars for the same problem. In HMMs, a tag is assigned to a word according to the tags of a limited number of neighboring words, but this may be naive from a linguistic point of view if the word's correct tag requires more distant relations [Tapanainen and Voutilainen, 1994]. The locality problem is more severe in inflective/agglutinative languages with free-word order [Hajic *et al.*, 2001]. Moreover, special care has to be taken if the number of tags is high which causes data sparseness problem in collecting statistics [Hakkani-Tür *et al.*, 2002]. The good thing about statistical approaches is that they always return a decision in every case of ambiguity, but the confidence of the correctness is not as high as in rule-based systems [Tapanainen and Voutilainen, 1994]. In studies that combine rule-based and statistical approaches, usually HMMs are used for the final decision [Ezeiza *et al.*, 1998], [Hajic *et al.*, 2001].

Turkish morphological disambiguation has been studied by rule-based, statistical and hybrid systems. A tagging tool was developed by Oflazer and Kuruöz [1994] in which disambiguation was based on local neighborhood constraints, heuristics and limited amount of statistics. Although a success rate of 98-99% on selected texts with minimal user intervention was reported, the authors confirmed that the system would perform worse on more substantial texts.

Oflazer and Tür [1996] proposed to combine hand-crafted rules and unsupervised learning in a constraint-based morphological disambiguation scheme. They obtained 96–97% recall with a corresponding 93–94% precision and an ambiguity of 1.02–1.03 parses.

A voting paradigm on constraint-based disambiguation was presented by Oflazer and Tür [1997] to overcome the rule-sequencing problem. Within that study, individual rules were used just to *vote* for matching parses, and at the end the analysis with the highest score is selected. 95–96% recall, 94–95% precision were reported with about 1.01 parses per token, which was better when compared to previous work [Oflazer and Tür, 1996].

Statistical morphological disambiguation of Turkish was performed by Hakkani-Tür *et al.* [2002]. Three different probability models based on trigram language statistics were proposed and tested. It was concluded that all the models performed better than the naive Bayes model (baseline tag model) and the best of those gave 93–94% accuracy with 1 parse per token ambiguity.

2.4 Pronunciation Ambiguities and Homograph Resolution

2.4.1 Resolution of Non-Standard Words

In most genres of text exist many words that are not ordinary. These *non-standard words* can be classified into dates, currencies, numbers, Roman numerals, fractions, abbreviations, e-mail, and web addresses. For text-to-speech systems, non-standart words introduce some problems in that the generation of their pronunciations differs from standard words greatly [Sproat *et al.*, 2001]. Most of the time a synthesizer needs to map a non-standard word to a sequence of standard words for a correct reading which is called text normalization [Sproat *et al.*, 2001, Olinsky and Black, 2000]. Another problem that arises on this conversion process is that the non-standard words have a higher tendency to be ambiguous than ordinary words [Sproat *et al.*, 2001] which cause various problems. Some of the ambiguities and pronunciation difficulties are as follows:

Date and Times: Dates and times may be expressed in many formats. These formats change from language to language, and most probably each language has more than one style of writing those stamps. **01.10.1999, 1/10/99, 01-10-1999, 1-Nov-1999, 1st November 1999** are some of the valid writings of a date in English. After normalization they are expected to be read as **first of November nineteen ninety nine**. Note that the format of the date stamp may change due to the language. For example in American English the format is 'Month-Day-Year' where in British English the sequence is 'Day-Month-Year'. The same date may correspond to **tenth of January nineteen ninety nine** in an American English synthesizer [Edgington *et al.*, 1996]. An ambiguity may occur if the year is omitted. **1/2** may be a date (first of February) or a fraction (one half).

Currencies: A currency token contains a currency symbol at the beginning or at the end, e.g **$10**, **1000YTL**. First the amount is read, and then the currency is pronounced in general as in **10 dollars**. There may be some abbreviations such as **$10K** which has to be resolved as **ten thousand dollars** or some exceptions such as **$10 billion** which must be normalized as not **ten dollars billion** but **ten billion dollars**.

Numbers, Roman Numerals, Fractions: The numbers, Roman numerals, and fractions are very frequently seen in texts. Context has an important role in resolution of those. **1986** may be read **nineteen eighty six** as a year quantifier or **one thousand nine hundred eighty six** as a cardinal number depending upon the context. Another frequent ambiguity is about the Roman numerals. The **I** token may have to be mapped to **one**, to **first** or to the first singular person preposition **I**. Fractions may have many different readings. The ratio **1/4** has three pronunciations in Turkish: **çeyrek**, **bir bölü dört**, and **dörtte bir**. Although the meaning

does not change much, the correct selection according to the context is an asset for a TTS system in that language.

Abbreviations and Acronyms: One of the most difficult tasks in a TTS system is the resolution of abbreviations and acronyms. **Dr.** in English may be *drive* or *doctor*, **St.** may be *saint* or *street*. An interesting (and maybe exaggerated) example of abbreviations is **He tried to walk on the Sun. Howard died.** [Edgington *et al.*, 1996]. It is not clear that Howard died because of an accident happened on the Sunday, or Howard died because of he walked on the sun of our solar system. Acronyms on the other hand are spelled, as in **IBM**, or read as it is an ordinary word, as in **NASA** in English [Sproat *et al.*, 2001, Mareuil and Floricic, 2001]. Note that for correct pronunciation of acronyms, the first step is to detect them in a given text. For the recognition of acronyms in a free text, Taghva and Gilbert [1999] introduces an automatic method based on inexact pattern matching algorithm. While reading as a normal word, special care must be taken that the standard pronunciation rules may fail for acronyms. For example, in Turkish **AIDS** is read as it is in English and does not obey Turkish pronunciation rules, and **RTÜK** is read as if the first letter is spelled and the rest is pronounced as a word. Mareuil and Floricic [2001] argued in his work that the rules of lexical stress assignment may change for acronyms on the pronunciation of Italian and French acronyms.

Among those well known examples of token types in a text that need normalization, are some more miscellaneous cases in real life. The absence of Turkish characters in e-mail addresses and URLs causes problems in the readings of those. As an example, the **cozumholding** in the web site name `www.cozumholding.com.tr` is pronounced as if it is **çözüm holding**. Non standard punctuations (****White** man can't jump**) and some humorous spellings (**goooooooood morning**) for emphasis are the other sources of problems in text normalization process.

The study of Sproat *et al.* [2001] is an excellent study on normalization of non-standard words. The proposed methodology involves a step by step procedure. The first task is the tokenization of the input text and detection of non-standard words. If a word could not be retrieved by a simple lexicon look up, then it is marked as a non-standard word. Note that although this works for English, for agglutinative and highly inflective languages such as Turkish or Finnish, a morphological analysis is required for the decision as those languages' have infinite number of word forms theoretically. During the recognition of non-standard words, one can also benefit from the dictionaries of common abbreviations or similar lists along with some handcrafted rules.

After the detection of non-standard words, some of them may need further splitting down for correct expansion, e.g. **Win2K** has to be split into **Win** and **2K** tokens for further steps. When such a splitting occurs, it must be remembered that

these tokens are grouped together [Edgington *et al.*, 1996], as these groupings are important in phrase detection while generating the prosody.

The most important part of the system is the classification of the extracted non-standard word tokens, by which appropriate tags indicating the type of the non-standard words are assigned to each. Special care has to be taken while deciding the taxonomy of the non-standard words. That classification must be general enough to cover all possible cases. Sproat *et al.* [2001] define three main categories for non-standard words as purely of alphabetical characters, containing digits or numerals, and miscellaneous ones. Each category has several subcategories. For example, the first category has an EXPN subtype which means that the non-standard word of that type should be expanded for correct reading, such as **gov't** be mapped to **government** and **N.Y.** to **New York**. The other subcategories of first are LSEQ, which means to read the non-standard word of that type as letter sequence (**IBM**), and ASWD, which dictates a reading as an ordinary word (**CAT**). After tokenization and splitting steps, each non-standard word token is put in one of those predefined classes. This classification scheme has been achieved via decision trees with the domain dependent and independent features extracted in the study of Sproat *et al.*[2001].

The tagged tokens are fed into the tag expansion step where the necessary normalization process is performed on each. This is the module where the mapping of non-standard words to standard words (*normalization*) is achieved. As all the words are ordinary after this step, the synthesizer can produce pronunciations.

There is one more problem to be solved: some tokens are ambiguous, because more than one tag may be appropriate or more than one expansion may be possible. Remember the examples that **11/2** may be *eleven over two* as a fraction or *eleventh February* as a date. **St.** may be *saint* or *street*. Those ambiguities are resolved by a language model which constitutes the last step of the system. This language model is based on n-grams, which is a way of disambiguation in local context.

The technique proposed by Sproat *et al.* [2001] is especially designed on English. The generality of the system is tested by deploying it on both Chinese and Japanese, which show quite different characteristics with the absence of word delimiters and high frequency of homographs observed in most texts [Olinsky and Black, 2000]. Olinsky and Black [2000] concluded that the system works also for these languages and so is a good basis of further studies on other languages as well.

2.4.2 Ordinary Words Requiring Sense Disambiguation

Apart from the difficulties faced with the pronunciations of non-standard words discussed in the previous section, some ordinary words need word sense disambiguation for correct reading. Such words, written identically, but read differently, may be classified into two cases according to their parts of speech tagging [Yarowsky, 1997].

The first case is of those words having different POS tags, e.g. the word **lives** has different pronunciations in sentences **Three lives were lost** versus **He lives in England**, where the word is a plural noun in the former and a verb in the second. The second case is when the words have the same part of speech tag and so the correct reading can only be achieved by semantic evidence. Note the word **bass** in the following sentences **John was playing bass guitar** vs. **New places for bass fishing can be found from internet** .

Generally, n-gram taggers, Bayesian classifiers, decision trees, and hybrid methods of those have been used in disambiguation [Yarowsky, 1997]. N-gram taggers use the sequence of POS tags around the target word and decide according to the n-gram statistics of POS tags previously collected. The technique is not capable of using long distance word associations, which is beneficial for the semantic disambiguation. As an example, the existence of words like **guitar**, **violin**, **percussion** is a very important information while disambiguating **bass**. Thus, although this method is efficient in the first case situations (different part-of-speech), it does not work for the second case (same part-of-speech), obviously.

Bayesian classifiers, on the other hand, have the ability to capture semantic evidence. Such classifiers begin by collecting a specific number of (e.g. 100) words near the target word. The order of the words is not considered. The total word association probabilities of those in the collection determines which pronunciation to be used. This approach does not benefit from local context information and also suffers from data sparseness problem.

Decision trees are complex rule based systems that perform disambiguation according to the features extracted. These features are basically orthographic, syntactic, and lexical properties [Hearst, 1991]. The disadvantages of using decision trees are the large parameter spaces and difficulties in construction of the decision rules. Yarowsky [1997] proposes a hybrid method with decision trees. The system depends on the measurement of collocations around the target word whose pronunciation is ambiguous. After the recognition of these phrases, the likelihood ratios are computed. That is mainly the probability of one possible pronunciation of the word given the existence of a collocation. The likelihoods are sorted into a decision list from where different models such as neural nets or Bayesian classifiers can be constructed for the final decision.

The discourse and collocations near a target word produce strong evidence to be used in sense disambiguation methods by the *one-sense per discourse* [Gale *et al.*, 1992] and *one-sense per collocation* [Yarowsky, 1993] properties of natural languages. These properties are summarized by Yarowsky [1995] as:

- One-Sense per Collocation: Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order and syntactic relationship.

- One-Sense per Discourse: The sense of a target word is highly consistent within any given document.

Although it is not specific for TTS systems, Yarowsky [1995] proposed an unsupervised learning algorithm based on these two properties. The algorithm begins with a small number of training seeds given for a polysemous word. For each sense of the target word, some samples are labeled in several texts. These are used as training seeds and by performing a classification scheme based on decision lists, the unlabeled occurrings are added to the set. This way the system learns by itself to differentiate between different senses of the target word. An accuracy rate of 96% with that bootstrapping procedure is reported.

Another recent work that benefits from both discourse and collocations on context sensitive homograph disambiguation for text-to-speech systems is proposed by Tesprasit *et al.* [2003] for Thai, which is also a language without word delimiters. The system uses two types of features: presence of *context words* within +/-10 neighborhood of the target and *collocations* that are up to 2 contiguous words around the target word. The technique is based on Winnow, which is a neuron-like network algorithm.

2.4.3 Named Entity Recognition

The words identifying special entities such as persons, organizations, and locations are very frequently seen in texts. In a TTS system these named entities cause problems in word pronunciation generation and phrase boundary detection for intonation.[7] Generating the correct pronunciations of those proper names is a great challenge because of the following reasons [Jannedy and Möbius, 1997] :

- The number of *named entities* is so large that it is not feasible to create a pronunciation lexicon that includes all. Although storing some of

[7] It is worth to note that the problems (and solutions to problems) of named entities heavily depends on the language of concern, meaning that the importance of the explanations may differ from language to language.

the most frequent ones in a dictionary seems a good idea, a robust system requires both recognition of known names and discovery of new names [Wacholder *et al.*, 1997].

- If we consider also the multilinguality of proper names, the situation worsens. Some of the foreign names have to be read in their original language pronunciation if they have not been linguistically assimilated in the phonological system. For example the reading of **McDonald's** in a Turkish TTS system has to be identical as in an English synthesizer.

- The grapheme-to-phoneme rules usually do not work for named entities. One may think to use the letter-to-sound rules of the language for named entities also, but most of the time the readings of proper names contradict with common phonological rules and the same grapheme strings may correspond to alternative pronunciations in different names (*idiosyncracy*).

- The lack of morphological analysis for proper names is also a source of problem as morphology serves a good basis for tagging and primary stress assignment processes explained in the previous section. If the morphological analyzer is not capable of decomposing the word, then it may have to undergo syllabification for pronunciation generation, where the grapheme-to-phoneme rules may not be appropriate for the correct reading. Note also that the semantic category of the proper name may be important in correct pronunciation generation. For example the Turkish word **'Aydın'** may refer two named entities. First, it may mean the city in Turkey as in the sentence **Aydın güzel bir ilimizdir.** (*Aydın is a nice city.*) and second a person name as in **Aydın dün okula gelmedi.** (*Aydın did not come to school yesterday.*). The stress assignments are different in both.

The second problem of named entities in speech synthesis is that the *structural ambiguity* Wacholder *et al.* [1997] introduced by the proper names may cause errors in phrase boundary detection, and as a consequence of that, the intonational patterns in a sentence may not be parsed correctly.

The structural ambiguities of named entities may be classified into three categories with respect to *prepositional phrase attachments*, *conjoined phrases*, and *possessive pronouns* [Wacholder *et al.*, 1997].

- Some of the proper names are formed in a way that two noun phrases are attached via a preposition (e.g. **"at"**, **"in"**, **"of"**). **"Midwest Center for Computer Research"**, **"City University of New York"**, **"The Museum of Modern Art in New York City"** may be given as examples of such.

Although those named entities are made up of the whole (all noun phrases plus all the prepositions), they may be erroneously divided into more phrases. For example, [**The Museum of Modern Art in New York City**] is a whole phrase referring a single entity, but it is hard to detect that it is not of the combination of three phrases, as [**The Museum**] **of** [**Modern Art**] **in** [**New York City**].

- Conjunctions of proper name phrases are another source of ambiguity. In the phrase **Barnes and Noble bookstore**, we know that **Barnes and Noble** is a single named entity and the phrase does not indicate two distinct companies as **Barnes** and **Noble**. Contrary to that, in the **Xerox and Bell laboratories** phrase, it is well known that **Xerox** and **Bell** are distinct corporations. A TTS system has to read [**Barnes and Noble**] as a whole phrase, where [**Xerox**] **and** [**Bell**] should be divided into two.

- The existence of possessive pronoun in a named entity has two meanings. First, it may be a relation between two as in **India's Gandhi**. Second, it may belong to the single proper name as in **Alzheimer's Association**. In the first case the items are disjoint (**[India's]** [**Gandhi]**), but in the second they form a single component (**[Alzheimer's Association]**).

Practically it seems best to have a dictionary of named entities along with their pronunciations while designing a full TTS system. If such a named entity dictionary exists, a recognizer of proper names must be able to benefit from that, but a mechanism to discover the new named items, which are missing in the lexicon, must also be implemented for a robust system.

Although the recognition and discovery of proper names in a text depend on the writing system of the language, generally the characteristics used in resolutions of named entities may be investigated in two parts as features coming from the word internal structure and features related with the context [Zhou and Su, 2002, Bikel *et al.*, 1997, Colins and Singer, 1999].

Word internal features are of those especially related with the spelling of the word. In most languages; the initials of proper names are capitalized. Note also that sentences begin with capital initial letters. Thus an ambiguity arises whether the first word of a sentence is a named entity or not. It is more confusing in some languages (such as English) if the named entity is the second word in the sentence where the first word is something like an adverb, a pronoun, or a preposition [Wacholder *et al.*, 1997]. For example in the sentence **New Zealand is near to Australia** it is hard to decide for a computer program whether the proper name is **New Zealand** or only **Zealand** only by examining only the initial letters of

words. The different word internal evidences, other than capital letters inside a named entity (*NameFinder*), may be stated as the inclusion of numbers (**Win2K**), punctuations (**N.Y.**), or special symbols (**AT&T**). Those evidences may be enlarged due to the structure of the language, or miserably may have no effect in named entity identification such as for languages of Chinese type.

Word-external or contextual features are found in the words near the proper names. A noun following a **Mr.**, **St.**, or followed by a **Ltd.** is most likely a proper name. The context sensitivity may be enlarged by looking up more distant words to the left or right around a named entity candidate, which is often a case to test especially for statistical recognition approaches [Lin and Hung, 2002]. Another word external feature may be questioning whether the candidate word has been previously labeled as a named entity.

With those types of features, the named entity recognition problem has been studied mainly in one of two ways: by hand-crafted rule based systems and by statistical methods. The named entity extraction tool, *Nominator*, of IBM is an example for rule based systems [Ravin and Wacholder, 1996, Wacholder *et al.*, 1997]. The Nominator does not benefit from any statistics or taggings and is based purely on heuristics. Following a proper name candidate search in a given text, it performs a splitting operation if a candidate is a multi-word token for the purpose of determining whether the conjunctions are within the named entity or not as described above. Then, it groups the detected names referring to same item (for example **JFK** and **John F. Kennedy** are put into same group). Last, a very important categorization is performed to classify if a named entity is a person, organization, or location. Note that from a TTS point of view, this is extremely important for some languages (see earlier examples of **Aydın** pronunciation in Turkish).

Similar to rule based systems, a finite state approach is given by Jannedy and Möbius [1997] for the pronunciation of proper names in German. The system is tested on street names of German where they mostly include `-dach`, `-stein`, `-hecke`, `-allee`, or `-platz`. The idea is to parse the given word using a finite state transducer searching these parts in the word. In case of success, the pronunciation is generated according to the predefined rules. If a failure occurs, the syllabification is deployed and the grapheme-to-phoneme conversion is performed. Note that special treatment has to be done in letter-to-sound conversion of proper names, and it is not the same model used in normal words of the language [Chung *et al.*, 2003].

The rule-based systems are hard to maintain and scale up. For each language, or maybe for each different domain, the rules have to be compiled differently [Lin and Hung, 2002]. The rules governing texts about drugs may not be the same as the rules compiled for texts about finance. Along the rapid development and robust structure of hand-crafted methods, statistical methods have been

preferred more recently. Note that statistical approaches require the calculation of some parameters for the models. These parameter estimations may be done in various ways including expectation maximization type bootstrappings or boostings [Lin and Hung, 2002, Cucerzan and Yarowsky, 1999, Collins, 2002].

One of the reasons for the wide usage of the statistical methods is that in some languages the features explained above are absent. In Chinese there is no white space word delimiter and no capitalization or similar evidences occurring in a given text. Thus, the Chinese named entity recognition problem is hard [Ye *et al.*, 2002, Lin and Hung, 2002]. A probabilistic verification method was proposed by Lin and Hung [2002] where the hypotheses of whether a candidate is proper or not is tested with a confidence measure. That confidence measure is defined to be a model looking at the right and left contexts of the candidate. If the returned value is above a threshold the candidate is assumed to be a proper name. A similar greedy strategy is also used by Ye *et al.* [2002] with a rationality model for named entity recognition in Chinese and both systems report success rates about 90%.

Machine learning based methods are also used in named entity recognition. An unsupervised classification scheme was proposed by Collins [2002] with an EM-style boosting and decision lists. With limited number of seed rules, the system is reported to be capable of learning the characteristics of proper names. A language independent system was supposed in Cucerzan and Yarowski [1999] which combines morphological and contextual evidences. Again a bootstrapping is performed with an initial training list of known proper names which are no more than a hundred items. The ranking algorithms can also be used for named entity recognition [Collins, 2002].

Among the numerous methods used in named entity recognition, the most compromising results are reported by the systems based on hidden markov models [Bikel *et al.*, 1997, Zhou and Su, 2002]. The *Nymble* [Bikel *et al.*, 1997] name finder treats each word as an n-tuple of features described above and calculates the required probabilities by just a simple counting on a previously labeled training set. The system puts each given word into one of seven proper name classes or into not-a-name class. 93% success was reported for Nymble in English. A more sophisticated HMM model was proposed by Zhou and Su [2002] which uses both deeper internal and external features of words. The success obtained is given as high as 96.6%.

2.5 Phrasing for Prosody Generation

TTS systems are now able to generate highly intelligible synthetic speech from unedited text input [Nooteboom, 1997], but they have some deficiencies in naturalness [Beutnagel *et al.*, 1999]. As the researchers aim to build synthesizers that

produce speech close to human speech as much as possible, more attention has to be paid for *prosody* generation. The prosody in a human speech depends mainly on two characteristics. The first one is the emotional situation of the reader while reading (for example if he is sad, angry, bored etc.) and the feelings that the read item gives to the reader (e.g., the subject may make the reader angry, sad, happy, etc.). Although it is possible to tune the parameters to make the machine read a text reflecting a given feeling, extracting those feelings or determining how the reader feels about what is being read from raw text is hard. Thus, this type of the prosody generation is more related with the behaviorial investigation of humans which is not discussed in this study.

The second type of prosody generation is dependent on syntax, semantics, and maybe pragmatics of the text. Many reviews state that earlier studies generally believed that syntax information is sufficient for prosody, but later understood that the syntactic structure only provides a good basis for prosodic structure, and the effect of the semantic and discourse also has great impact [Bachenko and Fitzpatrick, 1990, Wang and Hirschberg, 1991, Lindström *et al.*, 1996]. That is, syntax is necessary but not sufficient for prosody generation. Note that the prosody generation for more natural sounding needs higher levels of linguistic information.

In theory, the prosodic structure of a given utterance is a hierarchy of levels, from low to high : *syllable*, *foot*, *prosodic word*, *clitic group*, *phonological phrase*, *intonational phrase*, and *phonological utterance* [Nooteboom, 1997]. A foot is a lexical word which is made up of several syllables. If some of the consecutive foots in a sentence are pronounced together, they form a single prosodic word. A prosodic word may also be equal to just one foot if the pronunciation of that foot is not related with its neighbors. For example, in the sentence **I am going to the school** the foots **I** and **am** build up a prosodic (or phonetic) word and **going** is also a stand alone prosodic word and foot. A clitic group is a prosodic word plus the clitics to its right or left. In the previous example sentence **the** (clitic) plus **school** (prosodic word) is a clitic group. The phonological phrases are the concatenation of prosodic words (or clitic groups, if there exists any) and similarly intonational phrases are the union of those phonological phrases. The highest level is the phonological utterance which is the sentence that is the combination of intonational phrases.

In practice, the prosody generation process of a sentence begins with the words in it. Each word in the utterance has primary and maybe a secondary lexical stress in its syllables.[8] When a word is more intonationally prominent than others, as is the case in natural speech, the primary stress assigned syllable is accented and these

[8] The lexical stress assignment process is a word level issue related with the morphological analysis.

words are said to bear *pitch accent* [Hirschberg, 1993]. Although each word has a stress assigned syllable with the morphological analysis, the word is not accented all the time. When *deaccented*, no significant difference is observed between the syllables of the word. For example the word **object** has stress on its first syllable when used as a verb and in **I object your plan** it is accented, but in **They object it too** it is not [Hirschberg, 1993]. Additionally, a deaccented word may or may not be *cliticized*. When cliticized, the word is pronounced with its adjacent word as a one phonological word. As an example **to** is in cliticized form in **wanna** and not cliticized in **want to**.

The question of what makes a word to be accented or deaccented is not totally answered yet, but as mentioned before, syntactic, semantic and pragmatic properties are believed to effect accentuation. Hirschberg [1993] argued that surface order or some type of information status may be used in prediction of pitch accents. As an example for surface order, in the Turkish sentence **Bugün ben Ankara'ya gidiyorum** (*I am going to Ankara today*), **Ankara'ya** is accented, but in a different word order, such as **Ben Ankara'ya bugün gidiyorum**, **bugün** is accented. Many different information status phenomena may be defined, which effect accentuation, such as focus, contrastiveness, given/new distinction etc.. For example *given/new* distinction states if a content word in the sentence is introduced for the first time in the discourse (meaning 'new') or has been observed previously (meaning 'given'). If new, the word is accented, if not, it is deaccented, e.g., in sentence **There are lawyers and there are good lawyers** the first **lawyer** is accented, but the second is not.

Most text-to-speech systems perform accentuation based on content word/function word distinction. This approach seems easy and adequate for some right headed languages such as English but is not suitable for some languages such as Turkish. Liberman and Church [1992] names this approach as the *chinks 'n chunks* algorithm by calling function words as *chinks* and content words as *chunks*. The algorithm assumes that any phrase in a given sentence is of the form (chink*)(chunk). A phrase consists of all function words to the left of a content word. The parse of an example sentence with this algorithm is as [**There are several important changes**] [**in the way**] [**the quantifier rules**] [**will work**] [**for the remainder**] [**of the course**]. Note that a POS tagger is needed for the algorithm, which at least labels the words as chinks or chunks. On the other hand Hirschberg [1993] states that especially in synthesis of longer texts, this approach is problematic also for English. She argues that POS based parsing is not effective in complex nominals. For example **city hall** has the stress on the second word, **tax office** has on the first, and **city hall tax office** has on first and third, where all these phrases are noun-noun phrases. Those type of word groups are frequent in English, and special

processing for each type is costly. If someone wants to use chinks 'n chunks type algorithm, some rules to handle exceptions (or modifications) are necessary.

Detection of phonological phrases in a sentence is very crucial in prosody generation of a sentence. The syntactic parse of the utterance is generally assumed as a required information [Black and Taylor, 1994b, Bachenko and Fitzpatrick, 1990]. In their excellent study, Bachenko and Fitzpatrick [1990] state that syntax (syntactic categories of words and syntactic gaps between phrases) affects *phonetic quality*, *word prosody*, *phrasal stress*, and *segment quality*. The first two of those are investigated previously in this paper. For the remaining two, segment quality and phrasal stress, the following examples are given: For phrasal stress effect, **power units** has stress on **units** in the sentence **If house current fails, power units from battery**, where it has stress on **power** in **The power units failed**. The former is a verb-noun sequence and the later is a noun-noun sequence. For the segment quality, the word **to** is pronounced weak in **We spoke to John**, but in **Who did you speak to?**, it is read strongly because of the gap associated with question word **who**. Bachenko and Fitzpatrick [1990] begin with finding the phonological words at the first step for the detection of phonological phrases. At the second stage the formation of phonological phrases is performed via some rules that take syntactic constituents into account. Basically, a syntactic head (verb, noun, adjective, adverb) and the material that interneves between it and a preceding head is assumed to be a phonological phrase. After finding phonological phrase boundaries, some salience rules are applied to determine which of them are prominent. This may be viewed as the detection of intonational phrases in the theoretical structure discussed in the beginning of the section.

Head is central in the formation of phonological phrases in the study of [Bachenko and Fitzpatrick, 1990] and Lindström *et al.* [1996] proposed to use dependency graphs which are of the form head and modifiers and so seem appropriate more than the syntax trees. The idea is deployed on a Swedish text-to-speech system, where the output of a morphosyntactic component is used to build a dependency graph of utterances. The feasibility of the system is demonstrated by comparing the results with the human read sentences and it seems appropriate to use also dependency graphs in prosody generation.

There is not a direct mapping from syntactic parse of a sentence to its prosodic phrasing. A famous example of this is **This is [the cat that caught] [the rat that stole] [the cheese]**. The square brackets mark off the noun phrase constituents detected by syntactic analysis. This sentence is better synthesized as **[This is the cat] [that caught the rat] [that stole the cheese]**. Thus, although syntax is a good starting point, it is not always the true solution. In the same study, Bachenko and Fitzpatrick [1990] argued that the syntax/prosody misalignments

maybe a consequence of semantic considerations and readjustment rules. Taking those considerations into account is necessary for correct phonological parses.

Black and Taylor [1994b] propose a four step algorithm for the generation of prosody for an input sentence where each word in the given utterance has a POS tag, the syntactic constituent structure is given and also an *act* (yes/no question, statement, greeting etc.) is defined for the sentence. The first step, the prosodic phrase boundaries are detected in a similar work of Bachenko and Fitzpatrick [1990], and pitch accent information (accented, deaccented, etc.) for each word are assigned based to the work of Hirschberg [1993]. After converting the information of those two steps into a prosody tree in third step, the system uses the sentence act for the assignment of intonational boundaries. The rules related with the act of the sentence are deployed, and the resulting scheme defines the intonational phrases. By the use of that act information, different readings of the same sentence are possible, which is the case in real life. It is noteworthy here to state that practical examples of prosody generation for Italian are given by Ferri and Sanzone [1997] and a detailed work on generation of intonation in Swedish is defined by Horne and Filipson [1997].

Although the phonological phrases can be somehow extracted from syntax, intonational phrases are more difficult to detect [Atterer and Klein, 2002]. Atterer and Klein [2002] uses some form of chink 'n chunk algorithm for the detection of prosodic phrases in a German TTS system where a restructuring process of the chunks are performed via some rules. After the assignment of the prosodic phrases, intonational phrases are formed by first inserting breaks according to punctuation symbols. With the idea that the utterances are generally divided into equal length intonational phrases, further breaks are obtained by length and balancing constraints.

Different from the heuristics to specify intonations, Prevost and Steedman [1994] proposed an *information structure* which aims to synthesize speech from semantics and discourse of the text. Later, Prevost [1996] used this approach for spoken language generation. These type of synthesizers are named as concept-to-speech systems (CTT) to distinguish them from TTS systems which rely mostly on syntactic and orthographic information. It must be noted here that CTS systems also benefit from orthography and syntax, but build the main pronunciation scheme based on semantics. The proposed information structure is of two levels. In the first part, the utterances are divided into semantic propositions rather than syntactic constituents. Those propositions are either *themes* or *rhemes*. Basically, theme is the *topic* and rheme is the *comment*. In a wider sense, theme is the link by which the utterance is connected to previous sentences, and rheme is the core contribution of the utterance to the context [Prevost, 1996]. In the second part after the identification of themes and rhemes (which are actually the intonational phrases), the words that bear pitch accents in those segments are detected. This detection process is the

searching of emphasis among the words in every phrase. The given/new distinction and contrastiveness are used to recognize the focus words.

For a better understanding of the concept, the following examples are taken from the study of Prevost [1996]: In the sentence, **[The BRITISH amplifier produces][CLEAN treble.]**, the first square brackets enclose the theme, and the second marks the rheme of the sentence. The words written with upper case letters are the focus words to be accented. This sentence is likely to be the answer of the question **'What does the British amplifier produce?'**. The same sentence may be parsed as **[The BRITISH amplifier] [produces CLEAN treble.]** , where the first part is now the rheme, and the second part is the theme, if it is the answer of the question **What produces a clean treble?**. As can be seen from this example, a single utterance may be segmented into different themes and rhemes depending on the discourse. Also note that complex sentences may have more than one theme/rheme segmentation.

Rule based heuristics are used for the identification of themes and rhemes which is actually a difficult problem. Hiyakumoto *et al.* [1997] proposed a two level operation for this purpose. The algorithm begins by segmenting an input utterance into propositional constituents that are centered around verbs, adverbs, and prepositions (by the assumption that a POS tagging is initially provided). Each proposition should have only one verb complex. The punctuation symbols are also considered in this step. After the construction of those propositions, each proposition is subdivided into theme and rheme with some predetermined rules that are related with surface order or semantic interpretation of the verb constituent.

For the inspection of focus in themes/rhemes, givenness and contrastiveness metrics are used in the study of Hiyakumoto *et al.* [1997]. They benefit from the lexical database WordNet, where semantic relationships are encoded. In that database, each word is assigned to one of the four categories of noun, verb, adverb, or adjective. Each category has some number of synonym sets that are organized around semantic relations. As an example the nouns are connected with 'is-a', 'part-of', 'member-of', and 'made-of' relations.[9] The authors give well defined distinct algorithms for givenness identification and also for contrastive stress assignment. With these procedures, the focus items are detected in each theme/rheme segment. Although the system is just deployed on a small test set, the results are encouraging.

Intonational phrases not only affect naturalness in speech, but also have a great role in the meaning of a sentence sometimes. Wang and Hirschberg [1991] discuss the sample sentence **Bill doesn't drink because he's unhappy** to emphasize this. If uttered as a single intonational phrase, the sentence means Bill does indeed

[9] For detailed information on WordNet, please refer to `www.globalwordnet.org`.

drink, but the cause of his drinking is not his unhappiness. If it is read as two phrases, the interpretation changes that Bill does not drink and the reason for this is his unhappiness. In the same study, Wang and Hirschberg [1991] proposes a classification and regression tree (CART) analysis for the prediction of intonational phrasing from text. Based on some features, which include syntactic constituents and some other variables extracted from speech database of the corpus worked on, the system is trained, and 90% success is reported on the experimental test set. One of the main goals of the study in using CART analysis was to discover which features are relevant in automatic intonational phrase boundary detection.

Similarly, Sonntag and Portele [1997] have investigated the linguistic concepts effecting the prosody of spoken utterances. It is argued that the prosodic function of a sentence has to be separated into two as purely prosodic and segmental influences. The segmental influences are mainly the pauses occurring in speech which generally mark the intonational phrase boundaries. The authors aimed to investigate the factors that are independent of these pauses, and for this purpose the segmental information is removed from the test signals in the experiments of that study. In this way, although the intelligibility of the speech signal is reduced, all the information that is carried by prosody is made observable. Emotions, syntactic structure, dialogue acts (whether the sentence is an affirmation, negation, suggestion, or request) and given/new distinction are discussed as the main variables of prosody. Perception tests are performed to measure whether these variables can be detected from the signal. As an example, the following test is performed to investigate the effect of syntactic structure on pronunciation of a sentence: The subjects of the test hear of a speech signal which is the stimuli of a sentence whose segmental information is destroyed. The listeners are then asked to assign one of the given syntactic structures for the sentence heard. Approximately 70% of the time the listeners select the correct structure. Another test is made to measure the effect of dialogue act in prosody, where the listeners are asked of the act of a heard speech. For the details of the tests please refer to the original paper.

Another influence of accent in a sentence is the *anaphora* resolution. An anaphora is a reference to an item whose ambiguity is hard to resolve without discourse. Piwek [1997] discusses this subject on the example sentence: **John fed the animals. The cats were hungary.** If an accent is made on **the cats**, then it is understood that John fed many animals and among all the cats were hungary. Otherwise, it is the case that John just fed the animals and those animals are actually cats. A context sensitive disambiguation has to be performed to select between the first interpretation, which means `cats` are a subset of `animals`), and the second interpretation where the `animal` set is equal to the `cats`.

Chapter 3

THE PRONUNCIATION DISAMBIGUATION PROBLEM

Words typically have different pronunciations depending on their syntactic, and semantic properties in context. In Turkish, differences in pronunciation stem from differences in the phonemes used, the length of the vowel and the location of the primary stress [Oflazer and Inkelas, 2006]. The selection of the correct pronunciation requires a disambiguation process that needs to look at local morphosyntactic and semantic information to determine the correct pronunciation among alternatives. Disambiguating morphology serves a good starting basis for disambiguation of pronunciations, although it by itself, does not disambiguate all ambiguous cases of pronunciation. For example, determining the correct morphological analysis of the word **okuma** in Turkish, distinguishes between the possible pronunciations of this word in the sentences **'Okuma kitabı belirlendi.'** (*Reading book has been determined.*) and **'Saçma sapan şeyleri okuma.** (*Don't read those silly things.*) In the former, **okuma** is an infinitive form derived from verb **okumak** (*to read*) and corresponds to phonetic representation `/o-ku-"ma/` in SAMPA representation.[1] Note that `"` indicates the stressed syllable, and `-` indicates a syllable boundary. In the latter case the same word functions as an imperative form of the same verb, and pronunciation is represented with `/o-"ku-ma/` where the primary stress is on the second syllable. A text-to-speech system would have to take this into account for proper prosody.

Morphological analysis is a prior step to be performed in many natural language processing applications. A morphological analyzer produces all the possible morphological parses of an input word. Ambiguity arises if more than one analysis are generated for one word. For example, in a typical running Turkish text, every

[1] SAMPA(Speech Assessment Methods Pronunciation Alphabet) is an international machine-readable pronunciation alphabet. For further information, please refer to `www.phon.ucl.ac.uk/home/sampa`. See `http://www.phon.ucl.ac.uk/home/sampa/turkish.htm` for the set of Turkish SAMPA phoneme representations. We use the SAMPA notation to represent pronunciations in the text, where necessary.

word has on the average close to 2 morphological interpretations but about 60% actually have a single interpretation. Given a sequence of words, selecting the correct analysis among alternatives for each is defined as morphological disambiguation.

Let $W = w_1, w_2, \ldots, w_n$ be a sequence of `n` words. The set of morphological parses of w_i will be denoted by $M_i = \{m_{i,1}, m_{i,2}, \ldots m_{i,a_i}\}$, where a_i denotes number of distinct parses for w_i. Morphological disambiguation aims to select the correct $m_{i,j}$ for each w_i in the given context. Associated with each morphological analysis m_j, is one or more possible pronunciations of the word, s_k under that morphological interpretation.[2] Figure 3.1 illustrates the relation between the morphological analyses and pronunciations of words in a context.

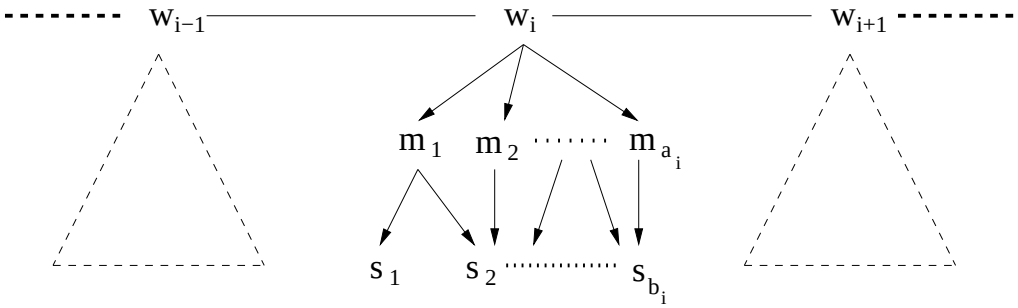

Figure 3.1: Pronunciations and morphological parses of words in a context.

As an example, Table 3.1, shows all morphological parses of word **karın** in Turkish with the corresponding pronunciations in SAMPA format, and the English gloss. There are five distinct morphological analyses but only three different pronunciations. Pronunciations 1 and 2 are associated with morphological parses 1 and 2, since the sense of the word is not in the morphological analysis. If the morphological disambiguation process for an occurrence of the word **karın** in a context results in m_3, the pronunciation would be s_3. Otherwise, if m_4 or m_5 is selected, then the reading is s_2. On the other hand, word sense disambiguation would be required to select the pronunciation in the cases of m_1 or m_2. Figure 3.2 represents the relationship between the morphological analyses and the corresponding pronunciation transcriptions of **karın**.

Text-to-speech systems have to generate the phonetic representations of the input words. Simple grapheme to phoneme conversions are not adequate for high quality output as orthography alone does not encode the phonological properties, such as stress. Various natural language processing techniques, which are investigated in literature survey section, are used in modern TTS systems for that purpose.

[2] We will avoid using multiples indices to denote morphological parses pronunciations and just refer to them with m_j and s_k respectively when the word index is obvious from the context.

Morphological Analysis	Pronunciation	Meaning
`kar+Noun+A3sg+P2sg+Nom` (m_1)	`/ca:-"r1n/` (s_1)	*your profit*
	`/ka-"r1n/` (s_2)	*your snow*
`kar+Noun+A3sg+Pnon+Gen` (m_2)	`/ca:-"r1n/` (s_1)	*of the profit*
	`/ka-"r1n/` (s_2)	*of the snow*
`kar+Verb+Pos+Imp+A2sg` (m_3)	`/"ka-r1n/` (s_3)	*mix it!*
`karı+Noun+A3sg+P2sg+Nom` (m_4)	`/ka-"r1n/` (s_2)	*your wife*
`karın+Noun+A3sg+Pnon+Nom` (m_5)	`/ka-"r1n/` (s_2)	*belly*

Table 3.1: Possible morphological parses and pronunciation transcriptions of the word **karın**

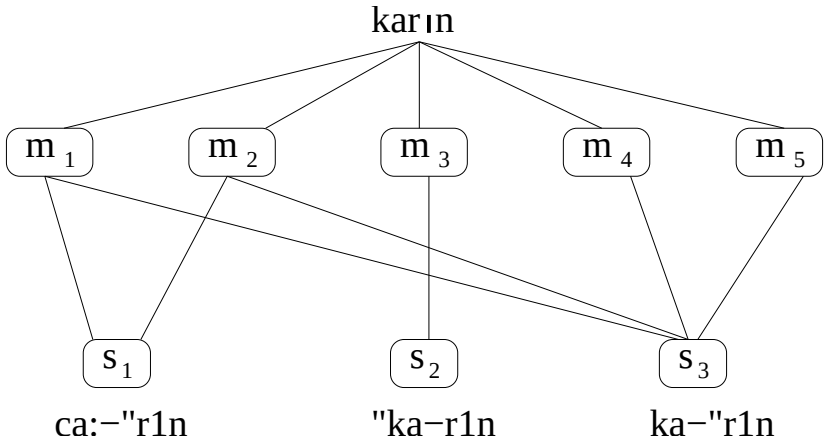

Figure 3.2: Graphical representation between the morphological parses and pronunciations of the word **karın**

Morphology serves a good starting basis for the basic phonetic transcription as the correct reading of a word heavily depends on its morphological properties. Every morphological analysis of a word corresponds to a pronunciation that some of those pronunciations are identical although the related analyses differ. That is because, in Turkish, distinct pronunciations per word are less than the distinct morphological parses.[3] If the correct morphosyntactic parse of a word is found, then most of the time the correct reading of that word is also detected. When the morphological disambiguator returns an incorrect result, it may still be the case that the corresponding pronunciation is the right one. Thus, the ambiguity among pronunciations of a word is a bit light when compared to that of the morphological parses. On the other hand, if two words are written in an identical orthography (homograph), but read differently according to their meanings (as in the case of m_1 and m_2 analyses of the example word), then *word sense disambiguation* has to be performed for the

[3] Average numbers of distinct morphological parses per token and distinct pronunciations per token are given as 1.86 and 1.11 respectively by Oflazer and Inkelas [2006].

generation of the correct phonetic transcription, which is obviously not required in the case of morphological disambiguation. In addition to that, the readings of named entities are another source of ambiguity and in some cases can not be solved by syntactic information. The relationship between the morphological disambiguation and pronunciation disambiguation problems may be sketched as in figure 3.3 hypothetically.

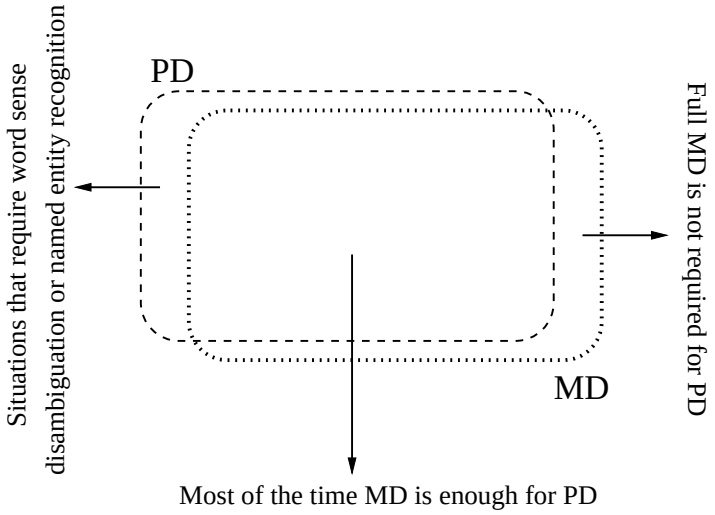

Figure 3.3: Comparison of the pronunciation disambiguation and morphological disambiguation problems

This dissertation aims to perform morphological disambiguation with application to pronunciation disambiguation in Turkish. The methodology of the study is applicable to other agglutinative and inflective languages as well.

Chapter 4

PRONUNCIATION AMBIGUITIES OBSERVED IN TURKISH AND DISAMBIGUATION TECHNIQUES

Turkish orthography uses 29 letters to encode its orthography, but phonologically there are 34 phonemes: the 8 vowels /i, y, e, 2, a, o, 1, u/ which correspond to *ı*, *ü*, *e*, *ö*, *a*, *o*, *ı*, and *u* in orthography and the 26 consonants: /p, t, tS, k, c, b, d, dZ, g, gj, f, s, S, v, w, z, Z, m, n, N, l, 5, r, j, h, G/. Orthography employs only 21 letters for consonants: /g/ and its palatal counterpart /gj/ are written as *g*, while /k/ and its palatal counterpart /c/ are written as *k*, /5/ and its palatal counterpart /l/ are written as *l*, /v, w/ are written as *v*, and /n/ and its nasal counterpart /N/ are written as *n*. Palatalized segments (/gj, c, l/) contrast with their nonpalatalized counterparts only in the vicinity of back vowels (thus *sol* is pronounced /so5/ when used to mean 'left' vs. /sol/ when used to mean 'note in scale'). In the neighborhood of front vowels, palatality is predictable (*lig* /ligj/ 'league').[1] /G/, written as *ğ*, represents the velar fricative or glide corresponding to the historical voiced velar fricative that was lost in Standard Turkish. When it is syllable-final, some speakers pronounce it as a glide and others just lengthen the preceding vowel. We treat it as a consonant for the purposes of this work and explicitly represent it. This inventory does not include long vowels – such phonemes are indicated with a vowel length symbol.[2]

The statistics given in Tables 4.1, 4.2, and 4.3 are taken from Oflazer and Inkelas [2006]. Table 4.1 compares the average numbers of morphological and phonological ambiguities per token and suggests their relationship.

There are three types of pronunciation ambiguities in Turkish arising from (i)

[1] In conservative spellings of some words, contrastive velar or lateral palatality is indicated with a circumflex on the adjacent vowel, though this convention actually ambiguous and because circumflexes are also used in some words, equally sporadically, to indicate vowel length.

[2] It is certainly possible to come up with a finer set of phonemes especially for text-to-speech purposes, so that the effects of palatal consonants, etc., can be distributed to the neighboring vowels.

Average Morphological Parse-Pronunciation Pairs / Token	1.86
Average Distinct Morphological Parses / Token	1.84
Average Distinct Pronunciations / Token	1.11
Average Distinct Pronunciations (ignoring stress) / Token	1.02

Table 4.1: Aggregate statistics over a 11,600,000 word corpus

N	% of Tokens with N Parse-Pron. Pairs	Cumul. % of Tokens with N Parse-Pron Pairs	% of Tokens with N Distinct Parses	Cumul. % of Tokens with N Distinct Parses
1	49.94	49.94	50.17	50.17
2	28.80	78.73	28.88	79.05
3	10.12	88.85	10.01	89.06
4	8.98	97.83	8.91	97.97
5	1.17	99.00	1.11	99.07
>5	0.99	100.00	0.92	100.00

Table 4.2: Distribution of parse-pronunciation pairs and parses

the phonemes used, (ii) the position of the primary stress, and (iii) differences in vowel length. The numbers in Table 4.1 indicate that the main source of ambiguity to be resolved in Turkish pronunciation is the position of the primary stress, and if we ignore the position of the stress, only 2% of the tokens have ambiguities such as differences in vowel length and consonant palatality in the root portions of the words.

It can be deduced from Table 4.3 that approximately 90% of the tokens have single pronunciation, 9% have two and only 1% have more than two distinct pronunciations.

N	% of Tokens with N Distinct Prons.	Cumul. % of Tokens with N Distinct Prons.	% of Tokens with N Distinct Prons. (No Stress)	Cumul. % of Tokens with N Distinct Prons. (No Stress)
1	90.08	90.08	98.32	98.32
2	9.37	99.45	1.68	100.00
3	0.52	99.97	0.00	100.00
4	0.03	100.00	0.00	100.00

Table 4.3: Distribution of pronunciation with and without stress marking

Differences in pronunciations manifest themselves in various combinations, and thus various techniques have to be applied to resolve the resulting ambiguities. These techniques are identified as: named entity recognition, word sense disambiguation,

and morphological disambiguation.

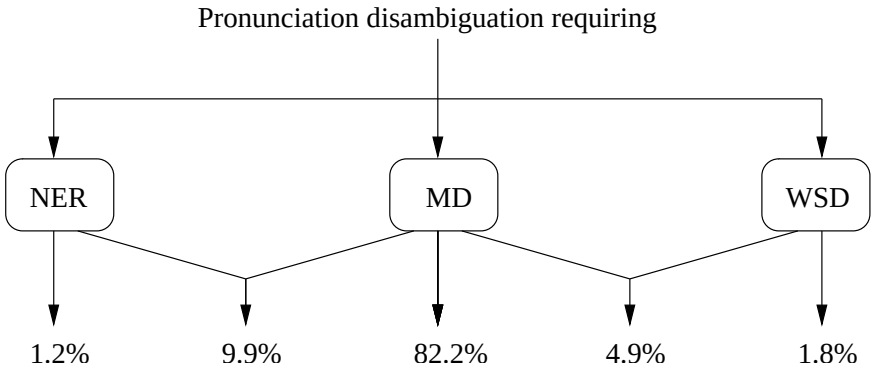

Figure 4.1: Pronunciation ambiguities classified according to the corresponding disambiguation methods

Figure 4.1 depicts the capabilities of each method to resolve ambiguous pronunciations. Although morphological disambiguation is sufficient most of the time to decide on the correct reading of words, sometimes named entity recognition and word sense disambiguation should also be used. For those words whose all possible parses are proper, named entity recognition is a must and those types occur 1.2% of the time. In 9.9% of the ambiguous pronunciations, some analyses of the word are proper and some are not. In this case morphological disambiguation and named entity recognition can be used in conjunction. If the morphological disambiguator decides on an analysis whose reading is unique, then there is no need for further investigation of named entity types. Otherwise, named entity recognition is required.

Word sense disambiguation should be used if all the parses are the same syntactically, which is observed in 1.8% of the case, and pronunciation differs according to the meaning of the word in the context. If not all the analyses are same, but there is only a subset of the parses that are equal, then morphological disambiguation may again be sufficient. That is when the unique analysis is chosen by the morphological disambiguator, then the problem is solved. Otherwise, word sense disambiguation is to be performed between the selected syntactically equal parses.

In Turkish, approximately 10% of words have more than one possible pronunciation. It can be deduced from figure 4.1 that in only 3% of the cases morphological disambiguation is not enough where named entity recognition and word sense disambiguation should be applied. On the remaining situations, morphological disambiguation either solves the pronunciation ambiguity exactly(82.2%) or helps for further steps (14.8%). The types of ambiguities are investigated with examples according to the required methodologies.

4.1 Pronunciation Ambiguities Solved by Morphological Disambiguation

If all the morphological analyses of a word are distinguishable from each other by their syntactic properties and each analysis has exactly one possible pronunciation, then morphological disambiguation resolves the pronunciation ambiguity. We demonstrate these cases with some examples below.

Homograph root words that have different parts-of-speech tags:

- **ama** (`/"a-ma/`, `ama+Conj`, *but*) vs. (`/a:-"ma:/`, `ama+Adj`, *blind*).
- **hala** (`/"ha:-la:/`, `hala+Adverb`, *still*) vs. (`/"ha-5a/`, `hala+Noun+A3sg+Pnon+Nom`, *father's sister*)
- **tabi** (`/ta:-"bi:/`, `tabi+Interj`, *certainly*) vs. (`/"ta:-bi:/`, `tabi+Adj`, *subordinate*)

The words are homographs but morphological analysis produces multiple segmentations giving rise to free and bound morphemes with different semantics, morphosyntactic functions and stress marking properties:

- **ajanda** (`/a-"Zan-da/`, `ajanda+Noun+...+Nom`, *agenda*) vs. **ajanda** (`/a-Zan-"da/`, `ajan+Noun...+Loc`, *on the agent*). Here the first parse has a root word with exceptional root stress.
- **fazla** (`/faz-"5a/`, `fazla+Adverb`, *much*) vs. **fazla** (`/"faz-5a/`, `faz+Noun+...+Ins`, *with the phase*). Here, the instrumental case marking morpheme (*-la*) is prestressing, but happens to surface as the last two phonemes of the first root word.
- **uyardı** (`/u-"jar-d1/`, `uy+Verb+...+Aor+Past+A3sg`, *s/he/it used to fit*) vs. **uyardı** (`/u-jar-"d1/`, `uyar+Verb+...+Past+A3sg`, *s/he warned*). In the first interpretation, the morpheme marking past tense is prestressing when preceded by the aorist aspect morpheme, but not otherwise.
- **attı** (`/"at-t1/`, `at+Noun+...^ DB+Verb+Past+A3sg`, *it was a horse*) vs. **attı** (`/at-"t1/`, `at+Verb+...+Past+A3sg`, *he threw*). Similar to above, in the first interpretation, the morpheme marking past tense is prestressing when applied to a noun or adjective root (through an implicit verbal derivation.)

4.2 Pronunciation Ambiguities Requiring Named Entity Recognition

A proper name may indicate different entities such as a city, a river, a person, and so on .... It may have different phonetic renderings according to the

entity it identifies. When the possible analyses of a word are all proper nouns, and there are multiple choices for its pronunciations, then named entity recognition should be used to detect the type, which will determine the correct phonetic representation. For example, the three morphological parses of word **Ilgaz** are all `Noun+Prop+A3sg+Pnon+Nom`, each of which indicates a small town in Turkey, a mountain chain or a person's name. The pronunciations corresponding to these are `/"15-gjaz/`, `/15-"gjaz/`, and `/15-"gaz/` respectively. As the morphological analysis is same for all, morphological disambiguation is useless here and one needs to perform named entity recognition to find out the correct pronunciation.

The readings of foreign proper names are another source of ambiguity. They may be read as in their original language or grapheme-to-phoneme rules may be applied to generate a suitable pronunciation. As an example, the phonetic renderings given by the pronunciation lexicon used in this study for **Hagi**, the name of a famous Romanian football player in Turkey, are `/ha-"dZi/` and `/ha-"gji/`.

The inflected forms of the proper names may have different phonetic transcriptions according to some morphological/phonological rules of the language. It has to be decided that whether the proper names will or will not obey these rules. For example, the last vowel of some root words drop when a suffix beginning with a vowel is attached. Related with this rule, possible parses and pronunciations of word **Anıtkabir'i** have two different possible readings as `/a-"n1t-ka-bi-ri/` and `/a-n1t-kab-"ri/`, if the rule applies or not, respectively.

4.3 Pronunciation Ambiguities Solved by Using Morphological Disambiguation and Named Entity Recognition in Conjunction

Proper nouns, especially those denoting place names that are homographs with common nouns (inflected or otherwise), usually have non-final stress in the root affecting the stress properties of their inflected versions: e.g., **Ordu**: (`/"or-du/` *name of a city*) versus (`/or-"du/`, *army*). (For example, *3. Ordu Futbol Şenliği* (*Third Ordu Soccer Festival*) vs. *3. Ordu Futbol Takımı* (*Third Army Soccer Team*).) Disambiguating whether a noun is a proper noun, by using orthographical cues such as initial capitalization and/or suffix separation characters, may not always be possible; one may have to use morphological disambiguation and named entity recognition techniques together. As an example, the word **Aydın** may correspond to a city in Turkey, a man's name, or an ordinary adjective meaning *bright* or *intellectual*. It has the pronunciation renderings `/"aj-d1n/`, `/aj-"d1n/` and `/aj-"d1n/`, respectively. If the result of a morphological disambiguation results that the word is used as an ordinary adjective in a given context, then the ambiguity is resolved. Otherwise, a named entity recognition task should be performed to find out the correct meaning

(a city or a person's name) which determines the correct pronunciation.

4.4 Pronunciation Ambiguities Solved Only by Word Sense Disambiguation

Pronunciation disambiguation of the words that have the same part-of-speech and inflect identically require word sense disambiguation. The word **adet** is such an example that word sense disambiguation should be used to determine the reading. It has two pronunciations as `/a-"det/` and `/a:-"det/` corresponding to meanings *piece* and *tradition* respectively. Part-of-speech tags of both are `Noun` and their all inflectional forms have exactly the same morphological analysis. Thus, it is not possible to disambiguate them using syntactic properties and word sense disambiguation should be applied to catch the correct sense, which also determines the correct reading, in a given context. This situation is similar to that of the English word **bass** which has two distinct interpretations, *fish* or *musical instrument*. Some other examples of that type ambiguity may be listed as;

- **şura**: (`/"Su-ra/`, *that place*) vs. (`/Su:-"ra:/`, *council*)
- **kar**: (`/"kar/`, *snow*) vs. (`/"car/`, *profit*)
- **yar**: (`/"jar/`, *ravine*) vs. (`/"ja:r/`, *lover*)

Note that since not so many such words exist in Turkish, selecting the most frequently occuring ones correctly solves the ambiguity most of the time.

4.5 Pronunciation Ambiguities Solved by Using Morphological Disambiguation and Word Sense Disambiguation in Conjunction

When some of the morphological analyses of a word are the same, and at least one parse is distinguishable from others by morphological properties, than morphological disambiguation is to be used first, and if inadequate, word sense disambiguation is further applied. The word **kola** is a good example of this process. The morphological parses, pronunciations and English gloss are:

`1.kola+Noun+A3sg+Pnon+Nom`, `/"ko-5a/`, *coke*
`2.kol +Noun+A3sg+Pnon+Dat`, `/ko-"5a/`, *towards the arm*
`3.kola+Noun+A3sg+Pnon+Nom`, `/ko-"5a/`, *starch*

Two of the analyses (1,3) are morphologically the same, but the root words possess different senses, and the remaining one (2) can be distinguished as it has a dative case marker. In a given context, the case tag of that word identifies the morphological analysis. If morphological disambiguation decides on the dative form,

then no further investigation is required as the second parse is unique. If this is not the case, nominative case marked parse is selected by the end of morphological disambiguation, then word sense disambiguation must be used to determine the correct pronunciation as there are two possible nominative candidates, each with distinct pronunciations.

Another similar situation occurs for word **hale**, which has the following analyses:

`1.hal+Noun+A3sg+Pnon+Dat`, `/ha-"le/`, *to the fruit market*
`2.hal+Noun+A3sg+Pnon+Dat`, `/ha:-"le/`, *to the state*
`3.hale+Noun+A3sg+Pnon+Nom`, `/ha:-"le/`, *halo*

The word **hal** may mean *fruit market* or *state*; both senses have the same pronunciation `/"hal/` when used in their root forms. The dative case marker suffixed to the word differentiates the first two analyses while the third analysis has the same pronunciation as does the second one. If the nominative case is selected as the result of the morphological disambiguation, word sense disambiguation is not required. Otherwise, word sense disambiguation must be applied to distinguish between the first and second analyses.

Chapter 5

STATISTICAL MORPHOLOGICAL DISAMBIGUATION BASED ON DISTINGUISHING TAG SETS

5.1 Modeling with Distinguishing Tag Sets

Turkish is an agglutinative language with a highly productive inflectional and derivational morphology. Morphosyntactic analyses of words in the language require a large number of tags to cover all the morphosyntactic and morphosemantic properties encoded in a word. The morphological analyzer used in this study has 116 tags, of which 28 mark the semantic properties, 15 are main parts-of-speech and remaining 73 represent the syntactic information.

Below is a sample syntactic analysis generated by the morphological analyzer for the word **hızlandırılmalıdır** (*it must be accelerated*):
`hız + Noun + A3sg + Pnon + Nomˆ DB + Verb + Acquireˆ DB + Verb + Causˆ DB + Verb + Pass + Pos + Neces + Cop + A3sg` [1]

Step-by-step derivations of the root word **hız** (*speed*) generates 4 inflectional groups listed below along with their semantic explanations:

1. `Noun + A3sg + Pnon + Nom`,
   (**hız**, *speed*)
2. `Verb + Acquire`
   (**hızlan**, *to speed up*)
3. `Verb + Caus`
   (**hızlandır**, *to make something speed up*)
4. `Verb + Pass + Pos + Neces + Cop + A3sg`
   (**hızlandırılmalıdır**, *to be accelerated by someone*)

Although the selected example is a bit exaggerated, it does not represent a rare situation. Similar long analyses are observed frequently in running text. The

[1] ˆDB mark derivational boundaries.

mean values for the number of tags and number of derivations observed in a syntactic parse are 4.27 and 1.34, respectively. Note that the 1.34 average number of derivation boundaries indicates that one third of the words have at least one derivation.

	Average
number of tags / morphological analysis	4.27
number of inflectional groups / morphological analysis	1.34

Table 5.1: Average numbers of tags and IGs per token

Table 5.2 contains the detailed statistics, collected from a one million-words corpus, showing distribution of the number of tags occurring in a morphological analysis. At most 23 tags are observed in a single parse.

N	% of tokens having N features in a morphological analysis
1	20.28
2	3.80
3	0.63
4	40.77
5	12.05
6	6.20
7	7.13
8	1.54
9	3.85
10	1.55
>10	2.2

Table 5.2: Distribution of the number of tags observed in morphological analyses of Turkish words

The morphological analyses can be split up to pieces named as inflectional groups (IGs) from derivation boundaries. Table 5.3 indicates the percentages of the number of IGs observed on morphological parses on the same corpus. 7 IGs are observed at most in an analysis.

Note that each IG is made up of a single part-of-speech (POS) tag, plus some others specifying the properties such as the agreement, case, tense, etc... Sometimes this may cause a confusion about which IGs' part-of-speech will represent the whole word. The POS tags of the example word given above are `Noun` for the first and fifth, `Adj` for the sixth and `Verb` for the rest of the IGs.

Statistical disambiguation methods that rely on n-gram statistics of all the tags suffer from the problem of data sparseness, since the tags set is very large. Hakkani-Tür *et al.* [2002] reported that number of possible IGs is 9129 while only

N	% of tokens having N IGs	Cumulative % of tokens having N IGs
1	74.02	74.02
2	18.87	92.89
3	6.13	99,02
>3	0.98	100

Table 5.3: Distribution of the number of inflectional groups observed in morphological analyses of Turkish words

2194 of them are observed on the corpus they used. By combining these facts, one can say that roughly there are 200K ($= 9129^{1.34}$) different types of morphological analyses. Then, the number gives a general idea as to the statistics required by a statistical disambiguater. It must also be taken into account that when an n-gram model is to be built with $n > 2$, the situation gets worse.

Thus, the challenge for languages that need large number of tags to fully cover their morphology is modeling morphological analyses with less fewer tags without losing necessary information to distinguish between parses.

Hakkani-Tür *et al.* [2002] have split up the morphological analyses across any derivational boundaries into inflectional groups (IGs) to partially overcome the problem and propose to model each morphological parse via these IGs. The best resulting model of that study represents each analysis by its final IG as a whole with all the tags it contains. Besides the final IG model, the disambiguation process additionally includes a separate root model.

Better handling of data sparseness can be achieved only if morphological parses can be represented by a smaller number of tags. Instead of using all the tags of the last IG as a whole, the main idea in this study is to use the discriminative subsets of these tags, which uniquely identify the analysis. This work proposes to represent each morphological analysis by such *distinguishing tag sets* of its final IG plus the part-of-speech of the root.[2]

The *distinguishing tag sets* (DTS) of a morphological parse are defined as follows:

Given a morphological analysis, let α represent the set of all subsets of the features used in its final IG. The distinguishing tag sets of the analysis are the

[2] This study is performed on the outputs of the Turkish morphological analyzer [Oflazer and Kuruöz, 1994], which assigns following tags as major part-of-speech to every IG: `+Noun`, `+Adj` (adjective), `+Adverb`, `+Verb`, `+Det` (determiner), `+Conj` (conjunction), `+Pron` (pronoun), `+Dup` (duplication), `+Interj` (interjection), `+Ques` (question), `+Postp` (postposition), `+Num` (number), `+Punc` (punctuation), `+BSTag` (beginning of sentence), `+ESTag` (end of sentence).

elements from α with the smallest size, so that if we determine that the correct analysis has those features, then we can uniquely identify this as the correct parse.

The main motivation is to be able to reduce the number of tags in the last IG as much as possible. Such a reduction results in a fewer number of distinct items representing the parses and thus instead of collecting statistics for thousands of feature sets, a few hundreds will suffice for the disambiguation process.

Another point is to eliminate the need for a surface root model. The root lexicon contains approximately 50K tokens, and a 3-gram root modeling of the language also suffers from data sparseness. The proposed methodology here does not include any root modeling. The POS of the first IG is concatenated to the DTS of the last IG to represent a syntactic analysis. Actually, the morphological disambiguator based on DTS performs the job by using less than 4 hundred such feature sets.

It is worthwhile here to explain the reason for selection of the root POS and DTS of the last IG together as the representative of morphological parses. Generally, the words of a sentence are syntactically connected to each other from their first and last IGs in Turkish. Most of the time, the intermediate derivations do not carry valuable information about the relationship between the words. The root POS is used to establish the connection to the previous ones and the inflectional features of the last IG link the current word to a subsequent head word. During the research conducted in this study, some different combinations of feature sets including {last POS}, {DTS of last IG}, {last POS, DTS of last IG}, { root POS, DTS of last IG} and {root POS, last POS} were explored with the observation that {root POS, DTS of last IG} is the best representative of a morphological parse. Thus, the proposed statistical morphological disambiguation is based on this modeling.

While finding the DTS of an analysis, the subsets of the last IG that have minimum cardinality are investigated according to the aim of using minimum number of tags. That is, first, we try to find if there are any single tags uniquely identifying the analysis among the other parses. If such a tag is detected, then it is assigned as DTS of that analysis. If not, the search continues with the possible combinations of 2 tags from the last IG, and so on. The pseudo code of finding the DTS of a given analysis is as follows:

```
DTS={};
STOP=false;
Remove any semantic tags occurring in the analysis;[3]
for i=1 to number of tags in the last IG of the analysis{
    G={all subsets of the tags in the last IG with i elements};
    for each subset S in G{
```

```
        if (the tags in S are all observed in the
                last IG of another analysis of the word){
                S_is_a_DTS = false;
        }else{
                S_is_a_DTS = true;
        }
        if (S_is_a_DTS==true){
                STOP = true;
                DTS=DTS ∪ S;
        }
    }
    if (STOP){
                break;
    }
}
```

Let us find out the DTS for each possible analysis of the word **çalışmaları**. It has the following parses:

1. `çalış+Verb+Pos^DB+Noun+Inf2+A3pl+P3sg+Nom`
2. `çalış+Verb+Pos^DB+Noun+Inf2+A3pl+Pnon+Acc`
3. `çalış+Verb+Pos^DB+Noun+Inf2+A3pl+P3pl+Nom`
4. `çalış+Verb+Pos^DB+Noun+Inf2+A3sg+P3pl+Nom`

For the first analysis, the last IG is `Noun + Inf2 + A3pl + P3sg + Nom`. At first, we search if any of these tags uniquely identifies the parse. We find that `Noun` and `Inf2` are observed in all the remaining analyses and since, they have no discriminative power. `A3pl` is also seen on the second and third, and similarly `Nom` on the third and fourth. The `P3sg` tag is only found on the first analysis and it is a DTS of the first analysis. Since a DTS is found with a single element, the stopping criteria holds and no further investigation of sets with more elements is performed.

For the second analysis, it is observed that `Noun`, `Inf2`, and `A3pl` from its last IG are not distinguishing tags as they exist on the other analyses as well. Either the `Pnon` or `Acc` identifies the parse separately. Thus, there are two distinct DTS of the second analysis: both {`Pnon`} and {`Acc`} are DTS's. Note that it is not necessary to have just a single item in the DTS of an analysis, and in this example there are two. While forming the DTS of the third analysis, none of the tags from its last

IG identifies the parse alone. The search continues with the combination of the two tags. Possible subsets of the last IG with two elements are generated as listed below:

1. {Noun,Inf2}
2. {Noun,A3pl}
3. {Noun,P3pl}
4. {Noun,Nom}
5. {Inf2,A3pl}
6. {Inf2,P3pl}
7. {Inf2,Nom}
8. {A3pl,P3pl}
9. {A3pl,Nom}
10. {P3pl,Nom}

Only the set number 8 among the 10 uniquely identifies the third morphological parse. The others appear in at least one of the remaining parses. The sets 1, 2, 3, 4, 5, 6, 7, 9 and 10 are also observed in parses (1,2,4), (1,2), 4, (1,4), (1,2), 4, (1,4), 1 and 4 respectively. Thus, the DTS of the third parse is a single item (8) formed by the combination of two tags.

The DTS of the fourth analysis is found to be the single A3sg tag as a result of a similar processing performed for the previous parses.

Parse #	*Possible DTS*	*POS of first IG*
1	{ P3sg }	Verb
2	{ Pnon }, { Acc }	Verb
3	{ A3pl,P3pl }	Verb
4	{ A3sg }	Verb

Table 5.4: Distinguishing tag sets of the morphological analyses of the word **çalışmaları** along with the POS of their first IGs'.

Table 5.4 summarizes the detected DTS of each analysis. The third column holds the POS of their first IGs and as all the parses of the word stem from a verbal root, the value is Verb. If we determine during disambiguation that the correct parse is {+P3sg} , then that is sufficient to deduce that the correct analysis is the first one. Similarly {+A3pl,+P3pl} and {+A3sg} imply the third and fourth

parses. There are two distinct DTS for the second morphological analysis and either {+Pnon} or {+Acc} identifies it.

Note that if no derivation boundaries exist in an analysis, then its first and final IG will be the same. In this situation the POS is also concatenated to the representation regardless of its existence in the DTS of the analysis. As an example Table 5.5 shows the DTS and POS tags of word **askeri**. Here, there is just one tag, Adj, on the third analysis that identifies the parse and also the POS of the parse. That analysis will be represented by (Adj,Adj) in the disambiguation process while the representatives of first and second are (P3sg,Noun),(Nom,Noun) and (Pnon,Noun),(Acc,Noun). There can be no duplicate elements in any of the

#	*Morphological Analysis*	*Corresponding DTS*	*POS of first IG*
1	asker+Noun+A3sg+P3sg+Nom	{P3sg}, {Nom}	Noun
2	asker+Noun+A3sg+Pnon+Acc	{Pnon}, {Acc}	Noun
3	askeri+Adj	{Adj}	Adj

Table 5.5: DTS investigation of word **askeri**, which means *his soldier*, *soldier (in accusative form)*, and *military* respectively.

generated DTS of an analysis, as the subsets of a set are unique. The problem arises if we cannot find any DTS after searching the last IG as a whole. Such a situation indicates that more than one analyses of the word has the same tags in their last IGs. In other words, the last IG of one parse is included in last IG of another parse. Generally this is observed when a word has parses as a proper name and also as an ordinary noun. **temel** is such a word and its syntactic analyses with their English gloss are :

1. Temel + Noun + Prop + A3sg + Pnon + Nom, *a person name*
2. temel + Noun + A3sg + Pnon + Nom, *the basis of a building*
3. temel + Adj, *fundamental*

Although we can differentiate the first parse from the others by the Prop tag, there are no distinct DTS for the second as Noun, A3sg, Pnon and Nom all exist in the last IG of the first. In such cases, the analyses whose DTS can be formed are processed normally. Thus, the DTS of the first is Prop and the third is identified by Adj. While searching the DTS for the problematic analyses, we do not attempt to distinguish them from the parses including such tags. Thus, when searching the DTS of the second parse, we will not try to distinguish it from the first one. The resulting DTS are Noun, A3sg, Pnon, Nom by this way. Note that the disambiguation process now can differentiate the first from second. If the disambiguator decides

on one of the `Noun`, `A3sg`, `Pnon` or `Nom`, the third analyses is eliminated from the possibilities list, but both the first and second stand for. Although the ambiguity is reduced, it cannot be totally resolved in such cases; further processing akin to named entity recognition, or word sense disambiguation is required. Those kinds of problematic situations have been seen 0.97% of the studied corpus.

N	Percentage of the analyses modeled with N tags
2	98.36
3	1.63
4	0.01

Table 5.6: Number of tags used in modeling of morphological parses via the proposed methodology

Table 5.6 lists the percentages of the number of tags used to model morphological parses via the proposed methodology. Remember that each morphological analysis is modeled with the distinguishing tags of its last IG with the part-of-speech tag of its first IG. It is worthwhile here to note that most of the time 1 tag from the last IG is enough to differentiate it from the others and only on very rare situations a combination of 3 is required.

Number of distinct DTS (K)	% of the analyses having K distinct DTS
1	53.47
2	14.22
3	2.49
4	20.56
5	9.14
6	0.12

Table 5.7: Distribution of the number of DTS for morphological analyses

Statistics on the number of distinct DTS generated for an analysis are also given in Table 5.7. A morphological parse may have more than one DTS, and, on the average, it has been observed that 2.6 distinct DTS are generated for each.

5.2 Morphological Disambiguation Based on DTS Modeling

Each morphological analysis $m_{i,j}$ is modeled by the DTS and the major part-of-speech tagof the first IG. Note that, a morphological parse may have more than one DTS, but only one root major POS. Following the notation used in statement of

the problem, for each $m_{i,j}$, let $O_{i,j}$ be the major POS tag of its first IG and $DTS^l_{i,j}$ be one of its DTS. The set

$$R_{i,j} = \{(O_{i,j}, DTS^1_{i,j}), (O_{i,j}, DTS^2_{i,j}), \ldots, (O_{i,j}, DTS^q_{i,j})\}$$

contains all distinct representations of $m_{i,j}$, assuming there are q different DTS identifying the analysis.

For the example word w_i **çalışmaları**, which was investigated in the previous section, the corresponding representative sets of its morphological parses are as follows:

1. $R_{i,1}$= {(Verb, P3sg)}
2. $R_{i,2}$= {(Verb, Pnon), (Verb, Acc)}
3. $R_{i,3}$= {(Verb, (A3pl,P3pl))}
4. $R_{i,4}$= {(Verb, A3sg)}

Remembering that a_i was the number of distinct analyses of word w_i,

$$R_i = R_{i,1} \cup R_{i,1} \cup \ldots \cup R_{i,a_i}$$

contains all representations of all morphological parses for w_i, where each element $(O_{i,j}, DTS^1_{i,j})$ of the set uniquely selects an analysis.

Finally, let us define t_i as the element selected from R_i by the morphological disambiguator. $t_i \in R_i$ determines the morphological parse. If $t_i \in R_{i,1}$ then $m_{i,1}$ is selected; if $t_i \in R_{i,2}$ then $m_{i,2}$ is selected. Similarly, $m_{i,3}$ and $m_{i,4}$ are selected if $t_i \in R_{i,3}$ and $t_i \in R_{i,4}$ respectively.

The disambiguation of a given sequence of words begins with the identification of $O_{i,j}$ and $DTS^l_{i,j}$ for all possible values of i,j, and l in the word sequence. A Hidden Markov Model is then constructed to compute for the sequence $T = t_1, t_2 \cdots t_i \cdots t_n$, where each t_i denotes a root POS and DTS combination referring to a unique morphological analysis $m_{i,j}$. The sequence T is computed in the usual way using Equation 5.1 by maximizing the probability $P(T \mid W)$:

$$\underset{T}{\operatorname{argmax}}\, P(T \mid W) = \underset{T}{\operatorname{argmax}}\, \frac{P(T) \times P(W \mid T)}{P(W)} \tag{5.1}$$

Since $P(W)$ is constant for every selection of T, Equation 5.1 becomes:

$$\underset{T}{\operatorname{argmax}}\, P(T \mid W) = \underset{T}{\operatorname{argmax}}\, P(T) \times P(W \mid T) \tag{5.2}$$

Turkish does not have morphological generation ambiguity so that the word formed by the given set of tags is unique.[4] Since each $t_i \in R_i$ determines the morphological parse uniquely, $P(W \mid T) = 1$ all the time, and hence the Equation 5.2 simplifies to:

$$\underset{T}{\operatorname{argmax}}\, P(T \mid W) = \underset{T}{\operatorname{argmax}}\, P(T) \tag{5.3}$$

The trigram approximation for P(T)

$$P(T) = \prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}) \tag{5.4}$$

can now be written as

$$P(T) = \prod_{i=1}^{n} P(\ (O_{i,x}, DTS_{i,x}^{l_i}) \mid (O_{i-1,y}, DTS_{i-1,y}^{l_{i-1}}),\ (O_{i-2,z}, DTS_{i-2,z}^{l_{i-2}})\ ) \tag{5.5}$$

where $t_i \in R_i$, $t_{i-1} \in R_{i-1}$, and $t_{i-2} \in R_{i-2}$; and x, y, z range over the respective number of ambiguous parses – $1 \leq x \leq a_i$, $1 \leq y \leq a_{i-1}$, $1 \leq z \leq a_{i-2}$.

The Viterbi algorithm can now be used on the expanded trigram model to find the highest scoring path, which makes up the sequence T.

As an example, Figure 5.2 illustrates the distinguishing tag modeling used in the morphological disambiguation of the utterance **Sadece doktora çalışmaları tartışıldı.** (*Only the Ph.D. studies were discussed.*). The morphological parses of the words along with their ($O_{i,j}$,$DTS_{i,j}^{l}$) pairs are as:

1. `sadece + Adverb` : $t_{1,1}$=(Adverb,Adverb)
   `sadece + Adj^DB + Adj + AsIf` : $t_{1,2}$=(Adj,Adj)[5]

2. `doktor + Noun + A3sg+ Pnon + Dat` : $t_{2,1}$=(Noun,Dat)
   `doktora + Noun + A3sg + Pnon + Nom` : $t_{2,2}$=(Noun,Nom)

3. `çalış + Verb + Pos^DB + Noun + Inf2 + A3pl + P3sg + Nom` : $t_{3,1}$=(Verb,P3sg)
   `çalış + Verb + Pos^DB + Noun + Inf2 + A3pl + Pnon + Acc` : $t_{3,2}$=(Verb,Pnon), $t_{3,3}$=(Verb,Acc)
   `çalış + Verb + Pos^DB + Noun + Inf2 + A3pl + P3pl + Nom` : $t_{3,4}$=(Verb,A3pl,P3pl)

[4] Refer to Hakkani-Tür *et al.* [Hakkani-Tür *et al.*, 2002] for a few rarely seen words that have this ambiguity.

[5] `+Asif` is a semantic marker and thus not included in the DTS generation.

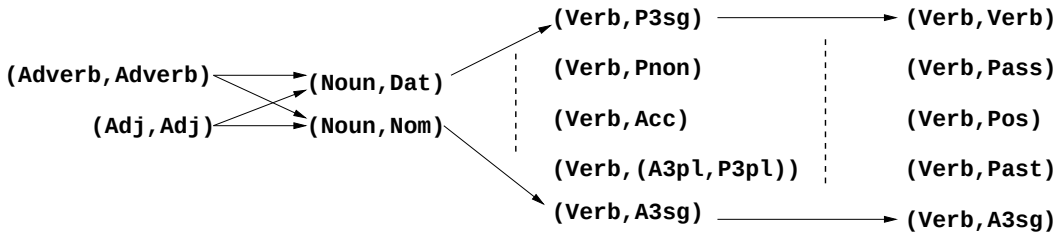

Figure 5.1: The sample sentence, **Sadece doktora çalışmaları tartışıldı.**, modeled with distinguishing tags

`çalış + Verb + Posˆ DB + Noun + Inf2 + A3sg + P3pl + Nom` : $t_{3,5}$=(Verb,A3sg)

4. `tartış + Verbˆ DB + Verb + Pass + Pos + Past + A3sg` : $t_{4,1}$=(Verb,Verb), $t_{4,2}$=(Verb,Pass), $t_{4,3}$=(Verb,Pos), $t_{4,4}$=(Verb,Past), $t_{4,5}$=(Verb,A3sg)

It should be noted that we can also use a formulation using *argmin* instead of *argmax* to find the worst scoring parses and remove them to *reduce* morphological ambiguity. This approach may also be meaningful in pronunciation disambiguation that does not need full-fledged morphological disambiguation. For example, the syntactic parses and pronunciations of word **kurdu** are given below with the corresponding meanings of *it was a court*, *assembled*, *his/her wolf*, and *wolf (in accusative form)* respectively.

1. `kur+Noun+...+Nomˆ DB+Verb+Zero+Past+A3sg`, `/"kur-du/`
2. `kur+Verb+Pos+Past+A3sg`, `/kur-"du/`
3. `kurt+Noun+A3sg+P3sg+Nom`, `/kur-"du/`
4. `kurt+Noun+A3sg+Pnon+Acc`, `/kur-"du/`

If the morphological disambiguator returns the first parse as the worst scoring analysis, the throw-worst approach eliminates it. Although syntactically the word will not be disambiguated at all, there will be no pronunciation ambiguity left since the remaining three analyses share the same pronunciation.

Both the *select-best* (using argmax) and *throw-worst* (using argmin) approaches have been implemented. It is observed that the throw-worst strategy gives better results for pronunciation disambiguation with a little penalty in the ambiguity ratio.[6]

[6] Please refer to the results and error analysis chapter for detailed information.

Chapter 6

IMPLEMENTATION

Experiments in this study were performed on the same 1 million-words corpus used by Hakkani-Tür *et al.* [2002]. This corpus was collected from a daily newspaper and contains approximately 50K sentences.

The morphological analyses of each word were generated by the Turkish morphological analyzer [Oflazer, 1994]. The current analyzer uses 116 feature tags of which 28 label semantic features. Throughout the study, those semantic tags were discarded and our model considered only the inflectional tags while computing the DTS. If two or more parses of a word differ only in semantic tags, then they are assumed to be the same analysis.

The pronunciations corresponding to morphological analyses are generated by a finite state pronunciation lexicon. The architecture and implementation of this pronunciation lexicon can be found in the study of Oflazer [2003].

The three evaluation metrics throughout the study are *precision*, *recall*, and *ambiguity*, whose definitions may be listed as:

$$\texttt{precision} = \frac{\texttt{number of tokens correctly disambiguated}}{\texttt{total number of parses}}$$

$$\texttt{recall} = \frac{\texttt{number of tokens correctly disambiguated}}{\texttt{total number of tokens}}$$

$$\texttt{ambiguity} = \frac{\texttt{number of parses}}{\texttt{number of tokens}}$$

Before any disambiguation, the precision was 55.9% and the morphological ambiguity was 1.8 parses per word. A preprocessing step which does not reduce the recall, was run on the test set before the actual disambiguator is run. The preprocessor performs some rule-based reductions to eliminate analyses that can only be seen in very restricted domains or are very infrequent or obsolete root words. The steps of this preprocessing are explained in the next section. At the end of this preprocessing stage, the morphological ambiguity becomes 1.45 while precision rises to 68%.

The morphological parses selected by the disambiguator by Hakkani-Tür *et al.* [2002] were used to train the statistical model using the CMU-Cambridge statistical language modeling toolkit [Clarkson and Rosenfeld, 1997]. As this training file also includes approximately 5% errors, some effort is given to enhance the training corpus by performing some error corrections. The corpus was split into 10 approximately equal pieces and a 10-fold cross validation scheme was used. In each of the tests, 9 of the segments were used as the training set from which the trigram statistics were extracted, and one segment was used as the test set.

6.1 Preprocessing Steps

The steps of preprocessing to lower the initial ambiguity ratio are improbable parse cleaning and rule-based reductions, including postpositional phrase disambiguation. Figure 6.1 depicts the precision and ambiguity after each step. Note that, while performing these reductions, special care has been taken not to sacrify the recall.

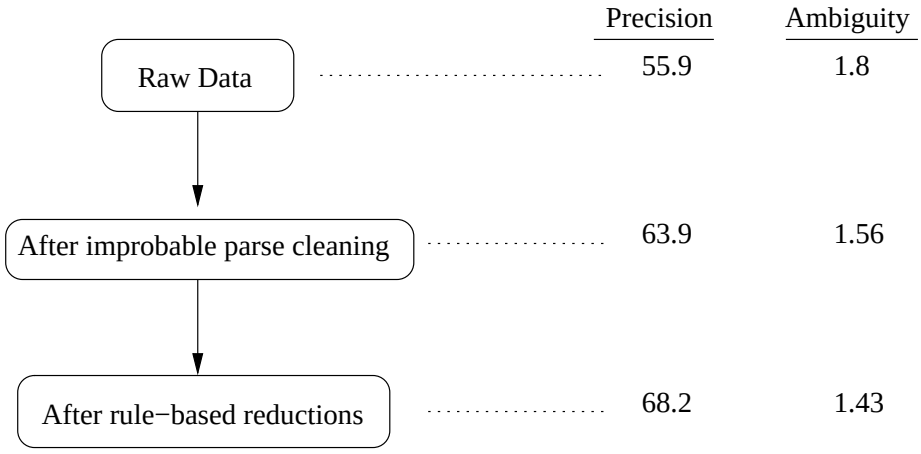

Figure 6.1: Precision and ambiguity ratios during preprocessing

Cleaning words with improbable parses eliminate analyses that can only be seen in very restricted domains or are very infrequent or obsolete root words. For example, although the root word **biz** is a pronoun indicating *we* nearly in all of its occurrences in a context, it can have an alternative noun analysis, which means *needle*. Thus, the `biz+Noun+...` type analyses can be ignored. Some other improbable parse examples are :

- `ada+Noun+...` (*my island*) parse generated for word **adam** (*man*),
- `on+Num+...` (*ten*) for word **onlar** (*they*),
- `altı+Num+...` (*your six*) for **altın** (*gold*)

Approximately 300 such cases have been detected and are cleaned from the morphological parses.

After cleaning the improbable parses, rule-based reductions are applied. The rules of these eliminations are listed below:

1. If the root words and the last IGs of two parses are the same, then one is eliminated. This situation occurs when there are more than one possible usages/meanings of the root word and they are derived to the same syntactic function by having their last IGs common. An example of such a word is **odur**, *that's it* or *that's he/she*. One of them is eliminated as arbitrary.

   1. `o+Pron+Demons+...+Nom^DB+Verb+Zero+Pres+Cop+A3sg`
   2. `o+Pron+Pers+...+Nom^DB+Verb+Zero+Pres+Cop+A3sg`

2. If a word has exactly two parses and one has `Noun+Zero+A3sg+Pnon+Nom` as its last IG, and both have `Num` as their root major POS, then the one with that last IG is eliminated. The reason is the lack of meaningfulness of the noun derivational forms of numbers in a given context. For example, the second analysis of the number **20** is to be eliminated below.

   1. `20+Num+Card`
   2. `20+Num+Card^DB+Noun+Zero+A3sg+Pnon+Nom`

3. If the roots of two morphological parses differ while their last IGs are same (or only differs in their semantic markers), and the longer root is a derived form of the shorter one, then the morphological parse corresponding to the short root is eliminated. The word **askerlik** is a good example to demonstrate the situation. The second analysis is eliminated as **askerlik** (*the military profession*) is a derived form of **asker** (*soldier*). Note that the only difference between the last IGs is the tag `Ness` which is a semantic marker and thus, neglected.

   1. `askerlik+Noun+A3sg+Pnon+Nom`
   2. `asker+Noun+A3sg+Pnon+Nom^DB+Noun+Ness+A3sg+Pnon+Nom`
   3. `asker+Noun+A3sg+Pnon+Nom^DB+Adj+FitFor`

4. The noun derivations of adjective root words that include the tag sequence `Adj^DB+Noun+Zero+A3sg` are eliminated. The third analysis of word **dışında** below is eliminated.

1. `dış+Noun+A3sg+P3sg+Loc`
2. `dış+Noun+A3sg+P2sg+Loc`
3. `dışı+Adj^DB+Noun+Zero+A3sg+P2sg+Loc`

5. If two of the analyses of a word are `Noun+A3sg+Pnon+Nom` and `Noun+A3sg+Pnon+Nom^DB+Adj+FitFor`, the derived adjective form is eliminated. For example, the following two analyses of word **sağlık** corresponds to meanings of *health* and *being right side* respectively, and the second one is eliminated, as such a usage is not seen in the language.

   1. `sağlık+Noun+A3sg+Pnon+Nom`
   2. `sağ+Noun+A3sg+Pnon+Nom^DB+Adj+FitFor`

6. The analyses `Verb^DB+Verb+Pass+Pos+Opt+A3sg` and `Verb+Pos+Opt+A3sg` are eliminated, if the word having these parses is not repeated consecutively in the given context. That is because the optative inflections of verbs are found as duplicate consecutive items generally. An example word **göre** means *according to*, unless it is repeated twice, which then corresponds to the optative inflection of verb *to see*, in the context.

   1. `gör+Verb+Pos+Opt+A3sg`
   2. `göre+Postp+PCDat`

The disambiguation of postpositional phrases is also performed as part of the preprocessing. Postpositions define a restriction on the case of the preceding word. Each morphological analysis of a postposition has a type tag from the set {`PCNom, PCDat, PCAbl, PCAcc, PCGen, PCIns`}, which specifies the case tag of the previous word. The case of a word should match with the type tag of the succeeding postposition; e.g. `PCAbl` identifies that the case of the preceding would be `Abl`. This constraint is a valuable information for disambiguation purposes. The examples of such disambiguations based on postpositional phrases are given below.

- The words in phrase **devletten yana** (*in support of the state*) has the following parses:

  1. `devlet+Noun+A3sg+Pnon+Abl`

1. `yan+Noun+A3sg+Pnon+Dat`
2. `yan+Verb+Pos+Opt+A3sg`
3. `yana+Postp+PCAbl`

As the preceding word is an unambiguous noun with ablative case, then it can be deduced that the correct morphological parse of the second word would be the third one.

- When a word has a possible postposition parse and none of the analyses of the preceding word agree with the type of that postposition, a reduction is achieved by eliminating the postposition parse, as postpositions always occur in phrasal structure. Given the sentence **"Maçın ikinci devresi çok çekişmeliydi."** (*The second period of the match was very contentious.*), the morphological analyses of **devresi** and **çok** are:

  1. `devre+Noun+A3sg+Pnon+Acc`
  2. `devre+Noun+A3sg+P3sg+Nom`

  1. `çok+Adj`
  2. `çok+Adverb`
  3. `çok+Postp+PCAbl`
  4. `çok+Det`

  As none of the parses of **devresi** has an ablative inflectional form, the third analysis of **çok** is irrelevant and can be eliminated.

The pseudo code given in Figure 6.2 is run when a word with an analysis including the `Postp` tag is observed. Note that the ambiguity reduction can be achieved both by selecting the correct parse or by throwing the illegal analyses.

Table 6.1 gives the percentages of the ambiguity reductions made by the preprocessing stages. These percentages are calculated by dividing the number of parses eliminated by each operation to the total number of eliminated morphological analyses. It is seen that removing improbable parses is 71.31% effective, while rule-based reductions and postpositional phrase disambiguations occupy and 24.66% 4.03% respectively.

The preprocessing of the training data is also done prior to extracting the necessary statistics.

Besides these operations, certain anomalies in the corpus were detected and corrected automatically. Beginning with high values of n, this methodology investigates if same n-gram phrases are differently resolved in various portions of the

```
Find the type case of the postposition;
If the word is ambiguous{
   If the previous word is ambiguous{
      If all the analyses of previous word include case type
         SELECT the postposition parse;
      else if none of the analyses of previous word include case type
         ELIMINATE the postposition parse;
   }else {
      If the analysis of the previous word includes the type case
         SELECT the postposition parse;
      else
         ELIMINATE the postposition parse;
   }
}else{
   If there is at least one previous parse having the type case
      ELIMINATE previous parses that do not include the type case;
}
```

Figure 6.2: The pseudo code executed when a word occurs with a postpositional parse.

Stage	Percentage of total reduction
Improbable Cleaning	71.31
Rule 1	2.94
Rule 2	5.08
Rule 3	11.58
Rule 4	3.54
Rule 5	0.77
Rule 6	0.75
Postpositional Phrase Disambiguation	4.03

Table 6.1: The percentages of each step at the preprocessing to reduce the initial ambiguity

corpora. The basic idea is as follows, for large n the disambiguated tags must be the same. If not, that means there exists a rarely seen situation which causes the statistical model to error. With this idea, all n-gram phrases for $20 \geq n \geq 4$ are explored and among the distinct ones occurring more than three times in the training file, if there are different resolutions of the phrase, the majority decision is made and the necessary corrections of morphological parses accomplished. Unfortunately, it is observed that not much correction is achieved. The results of this processing are summarized in Table 6.2.

n	*# of distinct n-grams occurring more than three times*	*# of corrections*
20	86	–
19	89	–
18	93	–
17	96	–
16	102	–
15	113	–
14	131	–
13	150	3
12	172	–
11	207	3
10	263	–
9	398	3
8	668	3
7	1105	8
6	2007	26
5	3907	137
4	8772	820
		Total: 1003

Table 6.2: Train file enhancement results by n-gram analysis.

6.2 System Architecture

The implementation of the proposed disambiguation on the corpus by 10-fold cross validation scheme is demonstrated in Figure 6.3. The training file, the enhanced output of the disambiguator by Hakkani-Tür *et al.* [2002], is divided into ten equal pieces, and each is disambiguated with the proposed model by using the language statistics extracted from the remaining nine folds.

The whole picture of the implemented disambiguation system is given in Figure 6.4. After the preprocessing operations described in the previous section, a named entity processing step is performed to disambiguate the proper names that can be identified by the orthographical cues such as initial capitalization and/or suffix separation characters. If the initial letter of a word, which is not seen at the beginning of the sentence, is capitalized, and it has morphological analyses including a `Prop` tag, then it can be deduced that the parses other than the ones with `Prop` tag can be eliminated. Note that unless a proper name exists at the beginning of a sentence, it can be identified by initial capitalization in Turkish. Although this ambiguity reduction may not resolve the analysis of named entity at all (if there exists more than one proper parse of the word), it reduces the ambiguity ratio. Approximately 70K proper names in the studied corpus are identified by this way.

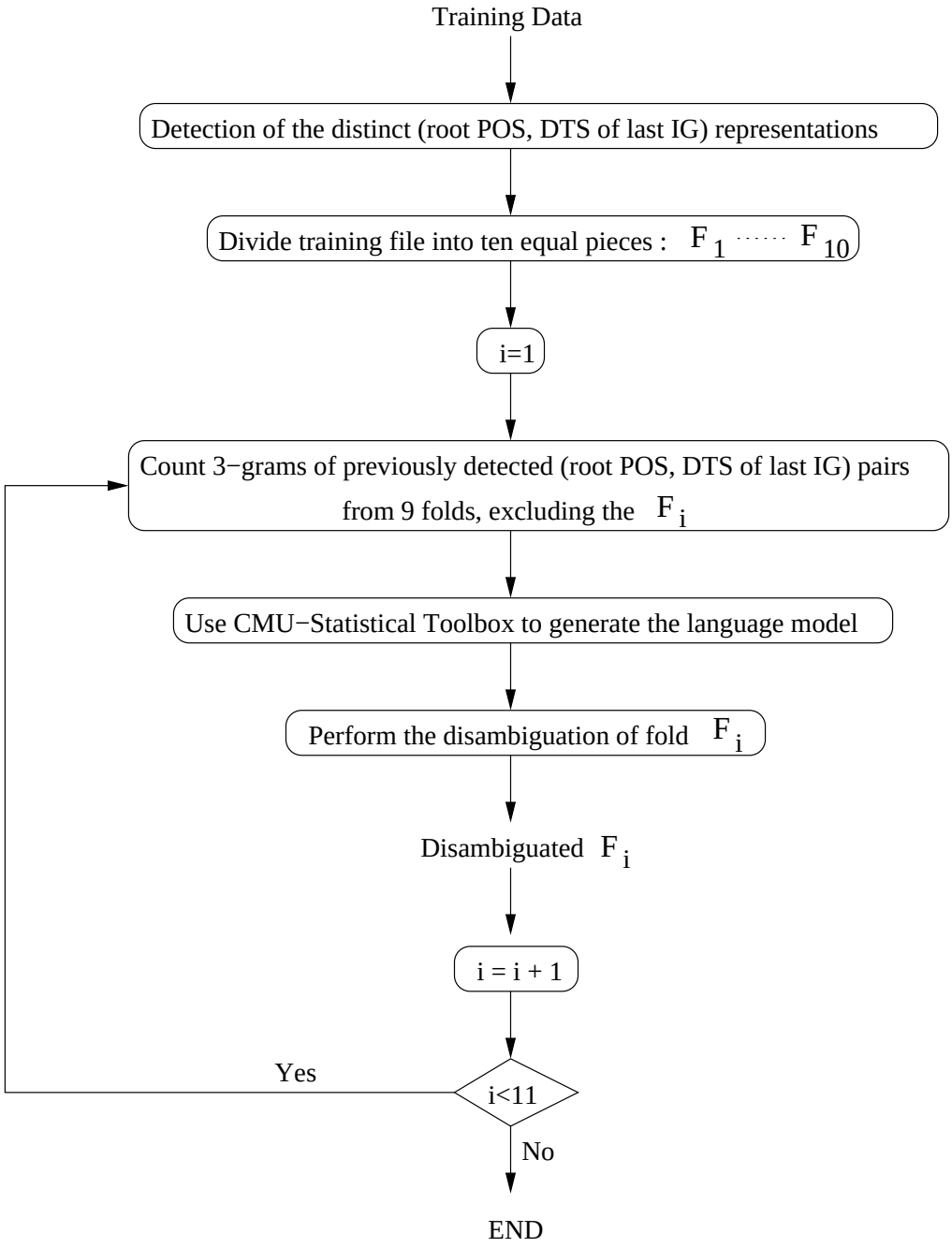

Figure 6.3: Implementation of 10-fold cross validation scheme

Statistical morphological disambiguation based on distinguishing tag sets is performed next. The operation is carried out in a sentence by sentence manner. Possible named entities are left with all their morphological analyses. In this way although the overall ambiguity is a bit increased, a specially designed named entity recognizer can be integrated to the system in the future to fully perform the named entity disambiguation processes that may also be required in generating the pronunciation renderings.[1] At the end of this stage the morphological disambiguation

[1] Refer to the related sections of pronunciation ambiguities chapter to remember

of the raw input text is finished.

Based on the outputs of the morphological disambiguation, the pronunciation renderings of the tokens can be selected. The pronunciation disambiguation problem may further require a word sense analysis operation for the situations described in the related sections of Chapter 4. Although word sense disambiguation is out of the scope of this study, a general heuristic has been applied, which selects the most frequent reading in case of such an ambiguity rising from the sense information of the word. Note that the designed disambiguator can be run with select-best or throw-worst criteria. If the aim is to generate pronunciations, it is better to use throw-worst. The results are investigated in the next chapter.

the details of these situations.

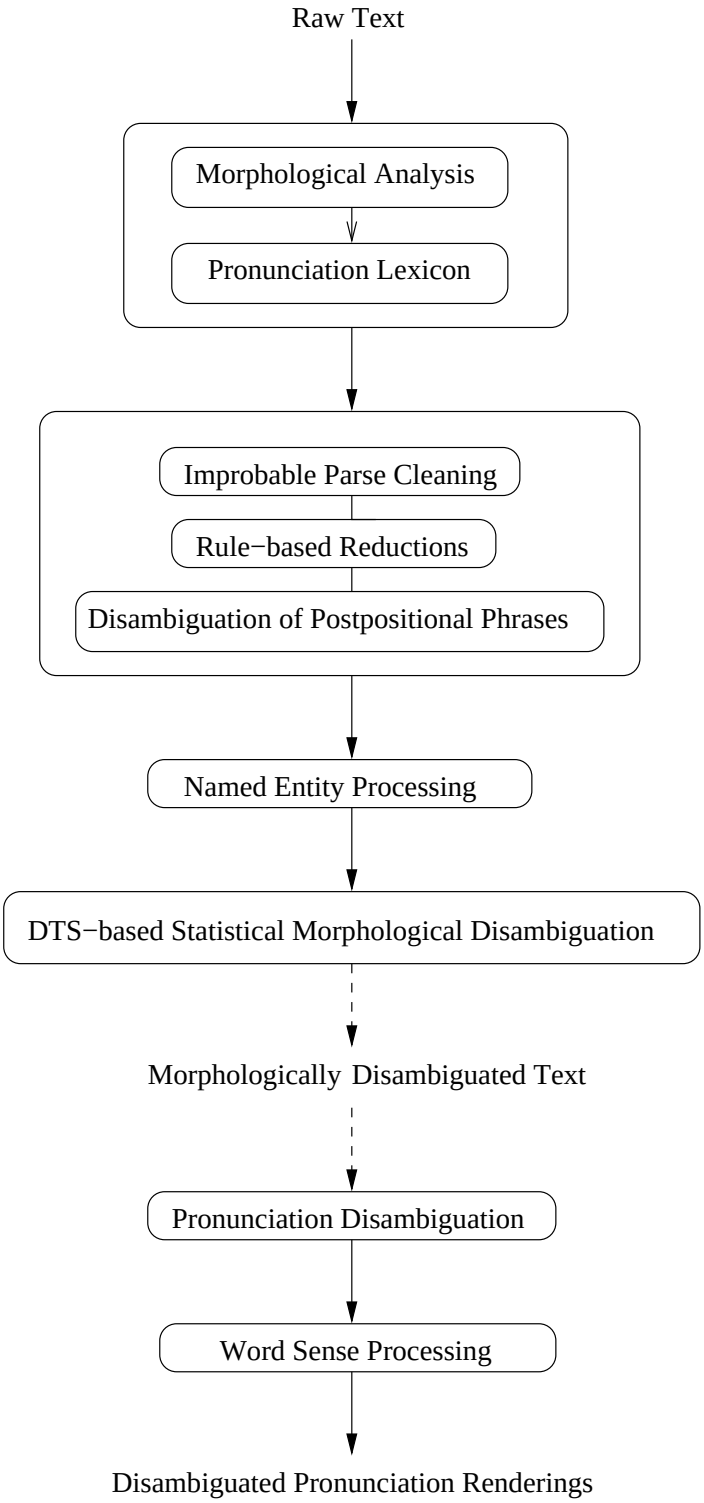

Figure 6.4: Overall system architecture

Chapter 7

RESULTS AND ERROR ANALYSIS

Table 7.1 offers the morphological disambiguation results of the implemented system with the 10-fold cross validation scheme. The highest recall obtained is 95.69% by the throw-worst strategy and 92.64% by select-best. The ambiguity is a bit higher and the precision lower in throw-worst strategy, as it operates by throwing the worst scoring analysis instead of selecting the best one. Note that proper names are not resolved at all within the system and left ambiguous.

Fold	SELECT-BEST			THROW-WORST		
ID	Precision	Recall	Ambiguity	Precision	Recall	Ambiguity
1	84.97	92.80	1.09	80.88	96.05	1.18
2	84.20	92.30	1.09	80.06	95.86	1.20
3	84.26	92.81	1.10	80.36	95.95	1.19
4	84.95	92.93	1.09	80.88	95.89	1.18
5	84.24	92.87	1.10	80.37	96.05	1.19
6	84.66	92.90	1.09	80.74	96.02	1.19
7	83.91	92.97	1.11	80.13	95.97	1.20
8	84.21	92.99	1.10	80.40	96.02	1.19
9	84.15	93.04	1.10	80.60	96.10	1.19
10	82.60	90.80	1.09	77.97	92.99	1.19
AVG	**84.22**	**92.64**	**1.10**	**80.24**	**95.69**	**1.19**

Table 7.1: Precision, recall, and ambiguity ratios of the implemented morphological disambiguator.

The disambiguation of pronunciations is achieved based on the morphological disambiguation. The results of each fold are stated in Table 7.2. The best scores are obtained with the throw-worst strategy, which is consistent with our initial guess of that full morphological disambiguation may be an overkill for pronunciation disambiguation. The corresponding precision, by being nearly 98%, is also very high.

It must be taken into consideration that although significant enhancements are performed on the training data, it still contains erroneous analyses and pronunciations. The parameters of the statistical model are extracted from this file. It

Fold ID	SELECT-BEST			THROW-WORST		
	Precision	Recall	Ambiguity	Precision	Recall	Ambiguity
1	98.28	99.13	1.01	98.06	99.58	1.01
2	98.29	99.10	1.01	97.96	99.53	1.02
3	98.23	99.17	1.01	97.96	99.55	1.02
4	98.28	99.09	1.01	98.02	99.55	1.01
5	98.04	99.16	1.01	97.72	99.54	1.01
6	98.11	99.10	1.01	97.75	99.49	1.02
7	98.13	99.25	1.01	97.84	99.60	1.02
8	98.12	99.22	1.01	98.70	99.58	1.02
9	97.90	99.11	1.01	97.64	99.57	1.02
10	98.11	99.01	1.01	97.81	99.45	1.02
AVG	**98.15**	**99.13**	**1.01**	**97.95**	**99.54**	**1.02**

Table 7.2: The results of pronunciation disambiguation

worths to note that the preparation of more accurate training files is a serious issue for future works.

On the investigation of errors, it has been observed that the root POS is found correctly in 68% of the errors, but the last IG observed only 8%. This implies, problematic cases occur because of the disagreement between the last IGs most of the time. A detailed look at erroneous results shows that the major POS of the last IG is also found correctly in 63% of the errors, which means the main source of the conflict occurs between the other subsidiary tags of the analyses. Observations on errors are summarized in Table 7.3.

Last POS correct:	63%
Root POS correct:	68%
Last IG correct:	8%
Root word correct:	70%

Table 7.3: Some observations on disambiguation errors

Another point is that the root words are found correct in 70% of the errors. As the erroneous cases are nearly 5% of the total, this implies the disambiguator selects the correct roots in 98.5%. The importance of this observation is that although the implemented system does not include any root language model, the proposed representations have the power to select the right roots.

A detailed investigation of the tags between the correct parses of the training file and the parses selected by the disambiguator indicate that the major disagreements occur between the number-person agreement tags {A1sg, A2sg, A3sg, A1pl, A2pl, A3pl}, and the possessive agreement tags {P1sg, P2sg, P3sg, P1pl, P2pl, P3pl, Pnon} with a ratio of more than 50% among the all conflicting features. These

types of errors especially originate from different possible analyses of the `'lAr+H'` (`leri,ları`) morpheme combination in Turkish. The possible parses of the word **görevlerinin** are given below with the corresponding English gloss to demonstrate the situation:

1. `görev+Noun+A3sg+P3pl+Gen`, *of theirs responsibility*
2. `görev+Noun+A3pl+P3pl+Gen`, *of their responsibilities*
3. `görev+Noun+A3pl+P3sg+Gen`, *of his/her responsibilities*
4. `görev+Noun+A3pl+P2sg+Gen`, *of your responsibilities*

All of these analyses can be observed in contexts having similar syntactic properties. As the statistical disambiguator decides on the correct parse according to the given context, actually it is expected that the system would make such mistakes. The remaining errors stem from the conflicts between various tags and some frequent examples of such are stated below to give a general intuition on potential erroneous situations:

dersleri :
1. `ders+Noun+A3sg+P3pl+Nom`, *their course*
2. `ders+Noun+A3pl+P3pl+Nom`, *their courses*
3. `ders+Noun+A3pl+Pnon+Acc`, *the courses (in accusative form)*
4. `ders+Noun+A3pl+P3sg+Nom`, *his/her courses*

çocuğun :
1. `çocuk+Noun+A3sg+P2sg+Nom`, *your child*
2. `çocuk+Noun+A3sg+Pnon+Gen`, *of the child*

kararı :
1. `karar+Noun+A3sg+P3sg+Nom`, *his decision*
2. `karar+Noun+A3sg+Pnon+Acc`, *the decision (in accusative form)*

yoksul :
1.`yoksul+Adj`, *poor*
2.`yoksul+Noun+A3sg+Pnon+Nom`, *destitution*

ancak :
1. `ancak+Conj`, *however*

2. `ancak+Adverb`, *hardly*

süren :

1. `sür+Verb+Pos^DB+Adj+PresPart`, *continuing*
2. `süre+Noun+A3sg+P2sg+Nom`, *your period*

tutacak :

1. `tut+Verb+Pos+Fut+A3sg`, *he/she will catch*
2. `tut+Verb+Pos^DB+Adj+FutPart+Pnon`, *something caught*
3. `tut+Verb+Pos^DB+Noun+FutPart+A3sg+Pnon+Nom`, *something caught*
4. `tutacak+Noun+A3sg+Pnon+Nom`, *pot-holder*

The errors observed on pronunciations relate with the errors of morphological disambiguation. Remembering the facts explained in chapter stating the pronunciation disambiguation problem, when the disambiguator makes a wrong selection, the corresponding pronunciation may or may not be erroneous. Nearly all the errors observed are caused by the morphological disambiguation, and there are very few examples stem from the word sense problems.

Chapter 8

A HEURISTIC ALGORITHM FOR PHONOLOGICAL PHRASE BOUNDARY DETECTION OF TURKISH

Phonetic words selected by the end of the pronunciation disambiguation process, include the position of the primary stress. Although this is enough for inner-word prosody, detection of the words that are to be accented or deaccented in a sentence with deeper syntactic and semantic analyses is to be performed for further inter-word prosodic events. While talking or reading, we, as humans, do not leave equal time intervals between the words. Phrase boundary detection is an important issue in synthesis of natural sounding speech to both adjust the durations between the tokens and to determine which ones to be accented/deaccented.

Most text-to-speech systems perform this boundary detection based on content word/function word distinction. This approach divides the words of a given utterance into two as content words and function words named as *chunks* and *chinks* respectively. The phrases are assumed to begin with a chunk and continue by any number of chinks [Liberman and Church, 1992]. For example, in sentence **"[She read] [the important pages] [in the park]"**, the words **the** and **in** are function words, thus, according to chinks and chunks algorithm, the phrases are marked between the brackets. Although this simple heuristic works fine on right headed languages such as English, it is not suitable for languages such as Turkish because of free word order structure, and also the difficulty in content/function word distinction, unclear in Turkish.

Dependency parsing was proposed as an alternative for phrase boundary detection [Lindström *et al.*, 1996]. Although it requires much deeper analysis than the simple chinks and chunks algorithm, this approach fits Turkish better. After the morphological disambiguation, prosodic phrases may be detected by applying simple dependency rules between the consecutive words. Here, it is not required to extract the whole dependency graph of a given utterance, but instead a light parsing is enough. Relations between the distant words are unimportant for prosodic structure since the aim is to locate phonologic groupings of neighboring words. Dependency parsing of Turkish has been studied with an extended finite state ap-

proach by Oflazer [2002]. Figure 8.1 demonstrates the relations between the words of a sample sentence from that work.

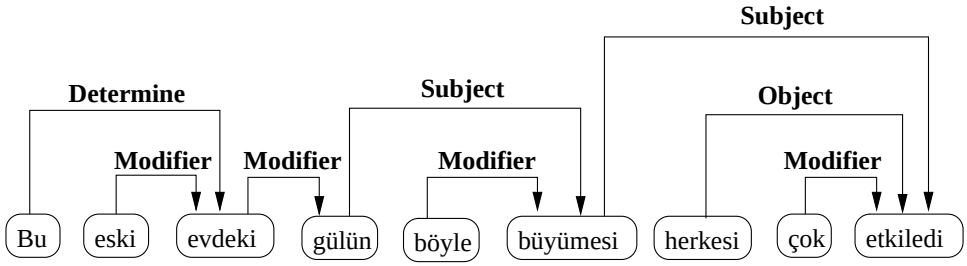

Figure 8.1: The dependency structure of a sample Turkish sentence.

On the example sentence, *subject*, *object*, and *determiner* relations are not between the consecutive words. Hence, they don't carry valuable information for phonological phrase detection. On the other side, although there is no link between the words **bu** and **eski**, they are in the same phrase. Thus, dependency parsing alone seems insufficient for phonological phrase boundary detection problem, and some extra rules must be compiled to process the prosodic interferences that are not handled in syntactic structure.

The following rules identify relations between consecutive words in a sentence which have been empirically constructed based on the noun phrase structure and dependency parsing of Turkish [Oflazer, 2002]:

1. A word whose POS of the last IG is `Adj`, `Det`, or `Num` is followed by a word whose last IG POS is `Noun`. This rule defines a syntactic relationship that the preceding token modifies the succeeding one. Some explanatory examples of such situations may be given as: **güzel**`(+Adj)` **ev**`(+Noun +A3sg +Pnon +Nom)`, *sweet home*; **birçok**`(+Det)` **araba**`(+Noun +A3sg +Pnon +Nom)`, *many cars*; **100**`(+Num +Card)` **dolar**`(+Noun +A3sg +Pnon +Nom)`, *hundred dollar*

2. Any number of consecutive words which have `Adj`, `Det`, `Num`, or `Adverb` tag in their last IG as major POS. This group of words forms a group of modifiers as in the phrase **bu eski evdeki** (*in this old house*), where **bu** is a determiner and **eski** is an adjective. Note that although dependency parsing does not link these words, from a phonological point of view, they are to be processed in the same phrase.

3. The word is a noun, pronoun, or postposition, followed by an adjective, adverb or noun which is derived from a verb root. This is another type of modify relation observed frequently in Turkish. For example, on the sample sentence given in figure 8.1, the words **böyle büyümesi** (`böyle +Adverb`

büyü +Verb +Posˆ DB +Noun +Inf2 +A3sg +P3sg +Nom, *such grow*) demonstrate the structure of this relation.

4. Postpositional phrases also constitute phonological phrases. A noun followed by a postposition forms such a phrase if their case tags agree. An example is : **başlangıcından beri** (başlangıç +Noun +A3sg +P3sg +Abl beri +Postp +PCAbl, *since its beginning*). Note that the case of the noun matches with that of the postposition by both being ablative (Abl and PCAbl respectively).

5. A noun in genitive or nominative case followed by another noun in any case constitutes a phonological phrase if the possessive agreement tag of the second one matches with the number/person agreement tag of the first noun. If the preceding noun is in nominative case and the succeeding noun includes the tag Pnon, meaning no possessive agreement, then the words are also linked. The phrase, **üyelerinin yerine**, (üye +Noun +A3sg +P3pl +Gen yer +Noun +A3sg +P3sg +Dat, *in place of its members*) is an example of a phonological phrase where the possessive tag of the second noun (P3sg) is compliant with the number/person agreement tag of the first one(A3sg). Other such examples are **görev süresi** (görev +Noun +A3sg +Pnon +Nom süre +Noun +A3sg +P3sg +Nom, *his/her service time*) and **çalıştığı ülkenin** (çalış +Verb +Posˆ DB +Noun +PastPart +A3sg +P3sg +Nom ülke +Noun +A3sg +Pnon +Gen, *of the country worked in*).

6. Similar to rule 5, if a noun in genitive or nominative case is followed by a derived adjective with Rel tag, a phonological link is to be established between them, when the possessive indicator tag of the IG before the last IG of adjective matches with the number/person agreement of the noun. An example is: **adanın kuzeyindeki** (ada +Noun +A3sg +Pnon +Gen kuzey +Noun +A3sg +P3sg +Locˆ DB +Adj +Rel, *in northern of the island*). Note that, most probably, the adjective further modifies the succeeding noun which will then constitute another phonological phrase.

7. A verb with a preceding noun in any case forms a phonological phrase. This is akin to subject/object relationships of dependency parsing. **hatırlatmak istiyorum** (hatırla +Verbˆ DB +Verb +Caus +Posˆ DB +Noun +Inf1 +A3sg +Pnon +Nom iste +Verb +Pos +Prog1 +A1sg, *I want to remind*) is a phrase exemplifying this rule.

8. A verb preceded by an adverb forms a phonological phrase. In this situation, the adverb modifies the verb. An example is:
şöyle anlattı (şöyle +Adverb anla +Verbˆ DB +Verb +Caus +Pos +Past +A3sg, *he/she explained such that...*)

For each word in a sentence, it is investigated whether there is a phonological link to the preceding or succeeding word conforming to one of the rules above. As each rule binds two words together, number of couples linked by each rule is given in Table 8.1 with the corresponding percentage. A word can be bound to both preceding and succeeding tokens by the defined rules. That constructs longer chains of words, the desired phonological phrases. Table 8.2 shows the word length of the detected phonological phrases. Note that in this Table, length 1 corresponds to the tokens that are not assigned to a phrase by the rules above. It is observed that the length of a detected phonological group is most of the time smaller than 5.

Rule #	# of pairs of words # matching the rule	Percentage of usage
1	82608	21.81%
2	29596	7.81%
3	61876	16.33%
4	13122	3.46%
5	133561	35.27%
6	1582	0.46%
7	48609	12.83%
8	7724	2.03%

Table 8.1: Frequencies of phonological link rules observed on the corpus.

Word length of the phrase	% Frequency of observation
1	57.26
2	21.61
3	11.12
4	5.42
5	2.44
>5	2.15

Table 8.2: Word length distribution of the detected phonological phrases.

The result of the defined phonological phrase boundary detection process on the example sentence **Milli Savunma Bakanlığı dövizli askerlik konusunda çözüm arayışına girdi.** (*The Ministry of Defense had begun searching for solutions for paid military service.*) is depicted below. Note that <PR*x*> and </PR*x*> mark the beginning and end of the applied number x phrase rule.

<PR5>
  <PR3>
    Milli `Milli+Noun+Prop+A3sg+Pnon+Nom`

Savunma savun+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Nom
</PR3>
Bakanlığı bakanlık+Noun+A3sg+Pnon+Acc
</PR5>
<PR5>
<PR1>
dövizli döviz+Noun+A3sg+Pnon+Nom^DB+Adj+With
askerlik askerlik+Noun+A3sg+Pnon+Nom
</PR1>
konusunda konu+Noun+A3sg+P3sg+Loc
</PR5>
<PR7>
<PR3>
çözüm çözüm+Noun+A3sg+Pnon+Nom
arayışına ara+Verb+Pos^DB+Noun+Inf3+A3sg+P3sg+Dat
</PR3>
girdi gir+Verb+Pos+Past+A3sg
</PR7>

Although it has been observed that nearly 98% of the time the length of the detected phonological phrases is lower than six words on the corpus, the phonologically linked list of words is occasionally so long that in reality no one can read that many tokens without a rest at some point. Thus, in the case of such long sequences of words combined together to form a phrase, some break points must be identified. Different heuristics can be proposed to handle those long sequences, such as partitioning into chunks smaller than five words, bisecting from the middle and maybe performing again some rule based operations. In this study, the phonological phrases longer than 5 words are handled by defining some pause locations inside the phrases based on the observation that people generally breathe at points of genitive nouns and participles (the words having `Prespart`, `Pastpart` or `Futpart` in their last IG) while reading or speaking. It must be noted here that these observations are purely subjective, since not much attention have been paid on both the detection and accentuation of the phonological phrases in Turkish.

In her book on Turkish phonology, Özsoy [2004] argues that the words that modify, determine, or somehow related to the head words are to be accented. She also notes that the speaker or reader specifies the important point of the utterance by the stressed word. For example, accenting the first word of the phrase **babamın yeni arabası** (*my father's new car*) emphasizes that the owner of the new car is the

father, while stressing the second word underlines that the car is the new one rather than the old one. Under normal conditions the second selection is more probable.

In their studies of Turkish stress assignment, Kabak and Vogel [2001], and Inkelas and Orgun [2003] argue that the leftmost accentable syllable is to be stressed in case of compound noun phrases. The intonation of noun compounds and genitive possessive noun phrases were explicitly explored in the studies of Levi [2002a], [2002b]. Although the number of sample structures investigated in her studies are rather limited, Levi discussed that the noun compound phrases have their first component promoted generally while the analysis of accentuation in genitive noun phrases vary. The experiments in her studies showed that the components of a genitive noun phrase may or may not retain their pitch accents. However the reason for that differentiation could not be totally identified. This dissertation promotes the first word of a genitive phrase if the second word begins with a vowel. That usage is based on the observation that people generally tend to read such phrases as a single lexical item in Turkish, thus promoting the word on the left of the phrase. If the second word is not beginning with a vowel than both words are promoted equally. With the proposed accentuation of genitive phrases, the first word of the phrase **babamın evi** (*my father's house*) is accented, while both of the words retain their pitch accents on **babamın sandalyesi** (*my father's chair*).

Based on these investigations and observations of Turkish phrasal stress, Table 8.3 depicts the component to be promoted by our previously explained rules that detect the phonological link between two consecutive words. The intonations of the phrases detected by the second rule (which connects consecutive modifiers or determiners) and the sixth rule (which is a special case of fifth rule) require their second token to be stressed more. The rest have their first words promoted. Only in some situations of the genitive noun phrases, both tokens retain their accent.

Rule #	Promote First Word	Promote Second Word
1	√	
2		√
3	√	
4	√	
5	√	
	√	√
6		√
7	√	
8	√	

Table 8.3: The accentuation table of the defined rules.

Initially all of the words in a given utterance are given zero intonation level.

After detecting the connected couples by the rules, the intonation levels of the linked words are increased according to Table 8.3. Each word may be linked to the preceding and succeeding one. This implies that the maximum level of intonation defined for any token may be at most 2. For example, while detecting the connections between the words in **sarı büyük kitap** (*yellow big book*), **büyük** is connected to **sarı** by the second rule and to **kitap** by the first rule. As second token is promoted by the second rule and first token by the first rule, the word **büyük** has an intonation level of 2.

Below is a sample sentence demonstrating the output of the phrasing and intonation level assignment of the whole system. The number written in bold between the braces at the end of each word indicates the level of intonation assigned for that word.
<PR5> <PR1> <PR2> <PR3> Kars'ta**(1)** yakalanan**(0)** </PR3> 500**(2)** </PR2> tüp**(1)** </PR1> zehirin**(0)** </PR5> <PR7> <PR3> <PR5> <PR1> <PR2> iki**(0)** milyar**(2)** </PR2> lira**(1)** </PR1> değerinde**(1)** </PR5> olduğu**(1)** </PR3> açıklandı**(0)** </PR7> (*It is stated that the 500 tubes of poison captured in Kars cost 2 billion Turkish liras.*)

Empirical observation performed on 100 sentences showed that approximately 85% of the time correct intonations are assigned to words. Note that the decision of correctness is subjective here.

Chapter 9

SUMMARY AND CONCLUSIONS

TTS synthesizers benefit from NLP techniques to generate more natural sounding speech [Külekci and Oflazer, 2004]. A survey of the language processing issues in speech synthesis has been given at the beginning of the study. While building a TTS engine that benefits from NLP, tokenization and vocalization are to be performed according to the orthography of the language prior to the morphological analysis. Possible speech transcriptions of each token are generated via a pronunciation lexicon based on the morphological analysis. Morphological disambiguation is a crucial step as in every NLP application. The results of the morphological disambiguator specifies also the correct pronunciations of the words. However, there may be some cases where named entity recognition or word sense disambiguation would be required. After the word level operations, phrase level events must be taken into consideration. Here the most important issue is the detection of the phrase boundaries, which again makes use of the syntactic analysis. By integrating these natural language processing issues, the prosodic structure of the input text is built and helps for the synthesizer to generate more natural sounding speech.

As stated, Turkish is an agglutinative language with a derivational morphology. Large number of tags are required to cover all morphological aspects of language. The current tag repository of the analyzer used in this study includes 116 features. On the average; each word has 1.84 distinct syntactic parses and each parse 4.27 tags. The problem of statistical morphological disambiguation of agglutinative languages suffer from data sparseness. To overcome this, morphological analyses of words should be modeled with a fewer number of tags. On the previous study of the statistical disambiguation of Turkish [Hakkani-Tür *et al.*, 2002], the authors propose representing each parse with its last inflectional group and report 2194 such feature sets on the test data. A separate root modeling of the language is also combined within that model.

With the aim of using minimum number of tags to identify a parse, distinguishing tag sets are introduced in our study. DTS are the minimum cardinality sets of tags from the last inflectional groups of morphological parses that uniquely identify

the analysis. Each morphological analysis is proposed to be represented by the DTS of its last IG combined with the part-of-speech tag of its root word. The statistical disambiguation of Turkish is achieved via just 374 such distinct representations on the same corpus with Hakkani-Tür. Note that the proposed disambiguator does not include a root modeling.

Some rule based reductions and cleaning of the very infrequent or obsolete analyses are performed to reduce the ambiguity ratio prior to the DTS-based disambiguation. Besides selecting the best sequence of analyses of words in a given context, the implementation has the ability to eliminate the worst scoring ones as an alternative approach. Although the ambiguity is a bit higher with that throw-worst strategy, recall values are better. The precision, recall and ambiguity ratios are 84.22%, 92.64% and 1.1 with select-best, and 80.24%, 95.69% and 1.19 with throw-worst strategies. Note that named entities are not forced to a single analysis by the system and thus, the ambiguity is not equal to one exactly in select-best experiment.

Throw-worst approach is especially thought to be ideal for pronunciation disambiguation application where a full morphological disambiguation may be an overkill. Turkish pronunciation ambiguities are discussed in detail within the study. The average pronunciations per token is 1.11. Most of the ambiguities of pronunciations can be resolved by morphological disambiguation. However, named entity recognition and words sense analysis may also be required in some certain situations, which are reviewed in the related chapter. Although full resolution of those situations is outside the scope of this study, they are handled by special processing based on some heuristic approaches. Orthographical cues are used to identify the proper names, and when more than one reading is possible for an analysis, the most frequent is selected. The disambiguation of pronunciations is achieved with a 99.54% recall ad 1.02 ambiguity ratio by the distinguishing tag based morphological disambiguator [Külekci and Oflazer, 2005] with the throw-worst strategy. This, to our knowledge, is the first attempt to disambiguate the pronunciations in Turkish.

Phrase boundary detection is an important issue in text-to-speech synthesis. Although the exact detection of phrases is not totally solved today, most TTS systems use heuristics based on function word / content word distinction to mark the boundary points. This approach does not work on Turkish because of the difficulty in discriminating between function and content words, and also the free word order structure of the sentences in the language. After morphological and phonological disambiguation steps, some effort has been given to investigate a heuristic to detect phonological phrases in Turkish. The rules, which are based on dependency parsing, have been constructed to explore phonological relationships between consecutive words. If there is such a relationship, the words are linked. The chains of these

links are proposed to constitute the phonological phrases.

Intonation is also an important point while generating natural sounding speech. Each rule is associated with an accentuation that defines which word of a couple is to be stressed more. Based on this empirical values, the words in a phonological phrase are assigned an intonation level. A subjective test performed on 100 sentences showed that 85% of the words are assigned reasonable levels.

It must be noted that there are not so many studies in the area of phrasal prosodic events of Turkish, and actually even the existing ones do not cover all the aspects to build a working system. Thus, while designing the heuristic and assigning intonations, empirical observations are taken into account. It is believed that deeper phonological analysis of the phrasal structures will led to better systems in practice. This attempt of phonological phrase boundary detection in Turkish may be applied to other languages which are not suitable for using function/content word distinction in phrase detection.

Bibliography

[Allen *et al.*, 1987] J. Allen, M.S. Hunnicutt, and D. Clatt. *From Text to Speech: The MITalk System*. Cambridge University Press, Cambridge, 1987.

[Atterer and Klein, 2002] M. Atterer and E. Klein. Integrating linguistic and performance-based constraints for assigning phrase breaks. In *Proceedings of COLING-2002*, Taipei, Taiwan, August 2002.

[Bachenko and Fitzpatrick, 1990] J. Bachenko and E. Fitzpatrick. A computational grammar of discourse-neutral prosodic phrasing in English. *Computational Linguistics*, 16(3):155–170, September 1990.

[Beesley, 1998] K.R. Beesley. Consonant spreading in Arabic stems. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the 36th annual meeting of ACL and 17th International Conference on Computational Linguistics*, pages 117–123, San Francisco, California, 1998. Morgan Kaufmann.

[Beutnagel *et al.*, 1999] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. The AT&T Next-Gen TTS system. Joint Meeting of ASA, EAA and DAGA, Berlin, Germany, available at `http://citeseer.nj.nec.com/beutnagel99nextgen.html`, March 1999.

[Bikel *et al.*, 1997] D.M. Bikel, S.M., R. Schwartz, and R. Weischedel. Nymble: A high performance learning name-finder. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, pages 194–201, Washington, DC., March 1997.

[Black and Lenzo, 2003] A.W. Black and K.A. Lenzo. *Building synthetic voices*. Language Technologies Institute, Carnegie Mellon University, `http://festvox.org/bsv/`, 2003.

[Black and Taylor, 1994a] A. W. Black and P.A. Taylor. CHATR: A generic speech synthesis system. In *Proceedings of the International Conference on Computational Linguistics COLING-94*, Kyoto, Japan, 1994.

[Black and Taylor, 1994b] A.W. Black and P. Taylor. Assigning intonation elements and prosodic phrasing for English speech synthesis from high level linguistic input. In *Proceedings of ICSLP'94*, volume 3, Yokohama, Japan, 1994.

[Brill, 1992] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of the third Applied Natural Language Processing*, Trento, Italy, 1992.

[Brill, 1995] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December 1995.

[Choueka and Neeman, 1995] Y. Choueka and Y. Neeman. Nakdan-text, an in-context text-vocalizer for modern Hebrew. BISFAI-95, 1995.

[Chung *et al.*, 2003] G. Chung, S. Seneff, and C. Wang. Automatic acquisition of names using speak and spell model in spoken dialogue systems. In *Proceedings of HLT-NAACL'2003*, pages 32–39, 2003.

[Church, 1985] K. Church. Stres assignment in letter to sound rules for speech synthesis. In *Proccedings of ACL'85*, pages 246–253, 1985.

[Church, 1986] K. Church. Morphological decomposition and stres assignment for speech synthesis. In *Proceedings of ACL'86*, pages 156–164, 1986.

[Clarkson and Rosenfeld, 1997] P. Clarkson and R. Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of Eurospeech'97*, pages 2707–2710, Rhodes, Greece, 1997.

[Colins and Singer, 1999] M. Colins and Y. Singer. Unsupervised methods for named entity classification. In *Proceedings of Joint SIGDAT Conference on Emprical Methods in Natural Language Processing*, pages 100–110, 1999.

[Collins, 2002] M. Collins. Ranking algorithms for named-entity extraction: Boosting and voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 489–496, Philadelphia, July 2002.

[Covington, 1993] M.A. Covington. *Natural Language Processing for Prolog Programmers*. Prentice Hall, 1993.

[Cucerzan and Yarowsky, 1999] S. Cucerzan and D. Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proceedings of Joint SIGDAT Conference on Emprical Methods in Natural Language Processing*, pages 90–99, 1999.

[Daelemans *et al.*, 1996] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. Mbt: A memory-based part of speech tagger generator. In *Proceedings of the Forth Workshop on Very Large Corpora*, pages 14–27, 1996.

[Edgington *et al.*, 1996] M. Edgington, A. Lowry, P. Jackson, A.P. Breen, and S. Minnis. Overview of current text-to-speech techniques: Part I - Text and linguistic analysis. *BT Technical Journal*, 14:68–83, 1996.

[Ezeiza *et al.*, 1998] N. Ezeiza, I. Alegria, J.M. Arriola, R. Urizar, and I. Aduriz. Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 379–384, Montreal, Quebec, Canada, 1998.

[Ferri *et al.*, 1997] G. Ferri, P. Pierucci, and D. Sanzone. A complete linguistic analysis for an Italian Text-to-Speech system. In J. Santen, J.P. Olive, R. Sproat, and J. Hirschberg, editors, *Progress in Speech Synthesis*, chapter 2, pages 123–137. Springer-Verlag, 1997.

[Fitt, 2001] S. Fitt. Morphological approaches for an English pronunciation lexicon. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, September 2001.

[Gal, 2002] Y. Gal. An HMM approach to vowel restoration in Arabic and Hebrew. In *Proceedings of ACL'02 Workshop on Computational Approaches to Semitic Languages*, University of Pennsylvania, Philadelphia, July 2002.

[Gale *et al.*, 1992] W.A. Gale, K.W. Church, and D. Yarowsky. One sense per discourse. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*, pages 233–237, 1992.

[Gou, 1997] J. Gou. Critical tokenization and its properties. *Computational Linguistics*, 23(4):569–596, 1997.

[Guo, 1998] J. Guo. One tokenization per source. In Christian Boitet and Pete Whitelock, editors, *Proceedings of the 36th annual meeting of ACL and 17th International Conference on Computational Linguistics*, pages 457–463, San Francisco, California, 1998. Morgan Kaufmann.

[Hajic and Hladka, 1997] J. Hajic and B. Hladka. Probablistic and rule-based tagger of an inflective language – a comparison. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Washington, DC, USA, March 1997.

[Hajic *et al.*, 2001] J. Hajic, P. Krbec, P. Kveton, K. Oliva, and V. Petkevic. Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of ACL'01*, Toulouse, France, July 2001.

[Hakkani-Tür *et al.*, 2000] D.Z. Hakkani-Tür, K. Oflazer, and G. Tür. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of COLING'2000*, volume 1, pages 285–291, 2000.

[Hakkani-Tür *et al.*, 2002] D.Z. Hakkani-Tür, K. Oflazer, and G. Tür. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36:381–410, 2002.

[Hearst, 1991] M. Hearst. Noun homograph disambiguation using local context in large text corpora. In *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, Oxford, October 1991.

[Heemskerk, 1993] J.S. Heemskerk. A probablistic context-free grammar for disambiguation in morphological parsing. In *Proceedings of EACL'93*, Utrecht, Nrtherlands, April 1993.

[Hirschberg, 1993] J. Hirschberg. Pitch accent in context: Predicting intonational prominence from text. *Artificial Intelligence*, 63(1–2):305–340, 1993.

[Hiyakumoto *et al.*, 1997] L. Hiyakumoto, S. Prevost, and J. Cassell. Semantic and discourse information for text-to-speech intonation. In *Proceedings of ACL'97, Concept to speech generation systems workshop*, pages 47–56, Madrid, Spain, July 1997.

[Horne and Filipson, 1997] M.A. Horne and M. Filipson. Computational extraction of lexico-grammatical information for generation of Swedish intonation. In J. Santen, J.P. Olive, R. Sproat, and J. Hirschberg, editors, *Progress in Speech Synthesis*, chapter 6, pages 443–457. Springer-Verlag, 1997.

[Inkelas and Orgun, 2003] S. Inkelas and C.O. Orgun. Turkish stress: A review. *Phonology*, (20), 2003.

[Jannedy and Möbius, 1997] S. Jannedy and B. Möbius. Name pronunciation in German text-to-speech synthesis. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, pages 49–56, Washington, DC., March 1997.

[Jilka and Syrdal, 2002] M. Jilka and A.K. Syrdal. The AT&T German text-to-speech system: Realistic linguistic description. In *Proceedings of ICSLP'2002*, September 2002.

[Kabak and Vogel, 2001] B. Kabak and I. Vogel. The phonological word and stress assignment in turkish. *Phonology*, (18):315–360, 2001.

[Kamir *et al.*, 2002] D. Kamir, N. Soreq, and Y. Neeman. A comprehensive NLP system for modern Arabic and modern Hebrew. In *Proceedings of ACL'02 Workshop on Computational Approaches to Semitic Languages*, July 2002.

[Karlsonn, 1990] F. Karlsonn. Constraint grammer as a framework for parsing running text. In *Proceedings of COLING'90*, volume 3, pages 168–173, Helsinki, Finland, 1990.

[Kontorovich and Lee, 2001] L. Kontorovich and D.D. Lee. Learning semitic vocalizations with hidden markov models. available at `www.cs.huji.ac.il/~oslkonto/hmmvocs.ps`, 2001.

[Külekci and Oflazer, 2004] M.O. Külekci and K. Oflazer. An overview of natural language processing techniques in text-to-speech systems. In *Proceedings of SIU'04*, Kuşadası,Turkey, April 2004.

[Külekci and Oflazer, 2005] M.O. Külekci and K. Oflazer. Pronunciation disambiguation in turkish. *Lecture Notes in Computer Science, ISCIS 2005 Proceedings*, 3733:636–645, 2005.

[Külekci and Özkan, 2001] M.O. Külekci and M. Özkan. Turkish word segmentation by using morphological analyzer. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, September 2001.

[Levi, 2002a] S.V. Levi. Limitations on tonal crowding in turkish intonation. In *Proceedings of 9th International Phonology Conference*, November 2002.

[Levi, 2002b] S.V. Levi. The realization of noun compounds and genitive possessive noun phrases. Technical report, University of Washington, 2002.

[Liberman and Church, 1992] M.Y. Liberman and K.W. Church. Text analysis and word pronunciation in text-to-speech synthesis. In S. Furui and M.M. Sondhi, editors, *Advances in Speech Signal Processing*, pages 791–831. Dekker, 1992.

[Lin and Hung, 2002] Y. Lin and P. Hung. Probabilistic named entity verification. In *Proceedings of COLING-2002*, Taipei, Taiwan, August 2002.

[Lindström *et al.*, 1996] A. Lindström, I. Bretan, and M. Ljungqvist. Prosody generation in text-to-speech conversion using dependency graphs. In *Proceedings of ICSLP'96*, volume 3, pages 1341–1345, Philadelphia, PA, USA, October 1996.

[Mareuil and Floricic, 2001] P.B. Mareuil and F. Floricic. On the pronunciation of acronyms in French and in Italian. In *Proceedings of Eurospeech 2001*, Aalborg, Denmark, September 2001.

[Nooteboom, 1997] S.G. Nooteboom. Text and prosody. In J. Santen, J.P. Olive, R. Sproat, and J. Hirschberg, editors, *Progress in Speech Synthesis*, chapter 6, pages 431–434. Springer-Verlag, 1997.

[Oflazer and Inkelas, 2003] K. Oflazer and S. Inkelas. A finite state pronunciation lexicon for Turkish. In *Proceedings of EACL workshop on finite state methods in NLP*, Budapest, Hungary, April 2003.

[Oflazer and Inkelas, 2006] K. Oflazer and S. Inkelas. The architecture and the implementation of a finite state pronunciation lexicon for Turkish. *Computer Speech and Language*, 2006.

[Oflazer and Kuruöz, 1994] K. Oflazer and İ. Kuruöz. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the 4th Applied Natural Language Processing Conference*, pages 144–149. ACL, October 1994.

[Oflazer and Tür, 1996] K. Oflazer and G. Tür. Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In E. Brill and K. Church, editors, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing*, 1996.

[Oflazer and Tür, 1997] K. Oflazer and G. Tür. Morphological disambiguation by voting constraints. In *Proceedings of ACL'97*, Madrid, Spain, 1997.

[Oflazer, 1994] K. Oflazer. Two level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148, 1994.

[Oflazer, 2002] K. Oflazer. Dependency parsing with an extended finite-state approach. *Computational Linguistics*, 2002.

[Olinsky and Black, 2000] C. Olinsky and A.W. Black. Non-standard word and homograph resolution for Asian language text analysis. In *Proceedings of ICSLP'2000*, Beijing, China, 2000.

[Özsoy, 2004] A.S. Özsoy. *Türkçe'nin Yapısı–I Sesbilim*. Boğaziçi University, 2004.

[Pfister and Romsdorfer, 2003] B. Pfister and H. Romsdorfer. Mixed-lingual text analysis for Polyglot TTS synthesis. In *Proceedings of the Eurospeech 2003*, 2003.

[Pfister, 1995] B. Pfister. The SVOX text-to-speech system. Computer Engineering and Networks Laboratory,Speech Processing Group, Swiss Federal Institute of Technology Zurich, available at `www.tik.ee.ethz.ch/~spr/publications/pfister:95d.ps`, September 1995.

[Piwek, 1997] P. Piwek. Accent interpretation, anaphora resolution and implicature derivation. In P. Dekker, M. Stokhof, and Y. Venema, editors, *Proceedings of 11th Amsterdam Colloquium*, pages 55–60, University of Amsterdam, ILLC/Department of Philosophy, 1997.

[Prevost and Steedman, 1994] S. Prevost and M. Steedman. Specifying intonation from context for speech synthesis. *Speech Communication*, 15(1–2):139–153, 1994.

[Prevost, 1996] S. Prevost. An information structural approach to spoken language generation. In *Proceedings of ACL'96*, pages 294–301, University of California, Santa Cruz, USA, June 1996.

[Ratnaparkhi, 1996] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Processing*, University of Pennsylvania, 1996.

[Ravin and Wacholder, 1996] Y. Ravin and N. Wacholder. IBM research report : Extracting names from natural-language text. Technical Report RC 20338, IBM T.J. Watson Research Center, 1996.

[Shalanova and Tucker, 2003] K. Shalanova and R. Tucker. South Asian languages in multilingual TTS-related database. Technical Report HPL–2003-9, HP Mobile and Media Systems Laboratory, Bristol, January 2003.

[Shih and Sproat, 1996] C. Shih and R. Sproat. Issues in text-to-speech conversion for Mandarin. *Computational Linguistics and Chinese Language Processing*, 1(1):37–86, August 1996.

[Sonntag and Portele, 1997] G.P. Sonntag and T. Portele. Looking for the presence of linguistic concepts in the prosody of spoken utterances. In *Proceedings of ACL'97, Concept to speech generation systems workshop*, pages 57–63, Madrid, Spain, July 1997.

[Sproat and Emerson, 2003] R. Sproat and T. Emerson. The first international Chinese word segmentation bakeoff. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 133–143, July 2003.

[Sproat *et al.*, 1996] R Sproat, C. Shih, W. Gale, and N. Chang. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404, 1996.

[Sproat *et al.*, 2001] R. Sproat, A.W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333, July 2001.

[Sproat, 1997] R. Sproat. Multilingual text analysis for text-to-speech synthesis. *Natural Language Engineering*, 2(4):369–380, 1997.

[Taghva and Gilbert, 1999] K. Taghva and J. Gilbert. Recognizing acronyms and their definitions. *International Journal on Document Analysis and Recognition*, 1:191–198, 1999.

[Tapanainen and Voutilainen, 1994] P. Tapanainen and A. Voutilainen. Tagging accurately – don't guess if you know. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, October 1994.

[Taylor *et al.*, 1998] P. Taylor, A.W. Black, and R. Caley. The architecture of the Festival speech synthesis system. In *Proceedings of Third ESCA/COCOSDA International Workshop on Speech Synthesis*, Australia, November 1998.

[Tesprasit *et al.*, 2003] V. Tesprasit, P. Charoenpornsawat, and V. Sornlertlamvanich. A context sensitive homograph disambiguation in Thai text-to-speech system. In *Proceedings of HLT-NAACL'2003*, pages 103–105, Canada, May 2003.

[Tür *et al.*, 1998] G. Tür, K. Oflazer, and N. Özkan. Tagging english by path voting constraints. In *Proceedings of COLING-ACL'98*, Montreal, Quebec, Canada, 1998.

[Voutilainen and Tapanainen, 1993] A. Voutilainen and P. Tapanainen. Ambiguity resolution in a reductionistic parser. In *Proceedings of EACL'93*, Utrecht, Holland, 1993.

[Wacholder *et al.*, 1997] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proceedings of Fifth Conference on Applied Natural Language Processing*, Washington, DC., March 1997.

[Wang and Hirschberg, 1991] M.Q. Wang and J. Hirschberg. Predicting intonational phrasing from text. In *Proceedings of ACL'91*, pages 285–292, University of California, Berkeley, California, June 17–21 1991.

[Webster and Kit, 1992] J.J. Webster and C. Kit. Tokenization as the initial phase in NLP. In *Proceedings of COLING-92*, volume 4, pages 1106–1110, 1992.

[Yarowsky, 1993] D. Yarowsky. One sense per collocation. In *Proceedings of the ARPA Human Language Technology Workshop*, 1993.

[Yarowsky, 1995] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

[Yarowsky, 1997] D. Yarowsky. Homograph disambiguation in text-to-speech synthesis. In J. Santen, J.P. Olive, R. Sproat, and J. Hirschberg, editors, *Progress in Speech Synthesis*, chapter 2, pages 157–172. Springer-Verlag, 1997.

[Ye *et al.*, 2002] S. Ye, T. Chua, and L. Jimin. An agent-based approach to Chinese named entity recognition. In *Proceedings of COLING-2002*, Taipei, Taiwan, August 2002.

[Zhou and Su, 2002] G. Zhou and J. Su. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, July 2002.