A STATISTICAL INFORMATION EXTRACTION SYSTEM FOR TURKISH

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BİLKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Gökhan Tür
August, 2000

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

Assoc. Prof. Kemal Oflazer (Advisor)

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

Prof. A. Enis Çetin

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

Asst. Prof. Bilge Say

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

Assoc. Prof. Özgür Ulusoy

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

Asst. Prof. İlyas Çicekli

Approved for the Institute of Engineering and Science:

Prof. Mehmet Baray
Director of the Institute

ABSTRACT

A STATISTICAL INFORMATION EXTRACTION SYSTEM FOR TURKISH

Gökhan Tür
Ph.D. in Computer Engineering
Supervisor: Assoc. Prof. Kemal Oflazer
August, 2000

This thesis presents the results of a study on information extraction from unrestricted Turkish text using statistical language processing methods. We have successfully applied statistical methods using both the lexical and morphological information to the following tasks:

- The *Turkish Text Deasciifier* task aims to convert the ASCII characters in a Turkish text, into the corresponding non-ASCII Turkish characters (i.e., "ü", "ö", "ç", "ş", "ğ", "ı", and their upper cases).
- The *Word Segmentation* task aims to detect word boundaries, given we have a sequence of characters, without space or punctuation.
- The *Vowel Restoration* task aims to restore the vowels of an input stream, whose vowels are deleted.
- The *Sentence Segmentation* task aims to divide a stream of text or speech into grammatical sentences. Given a sequence of (written or spoken) words, the aim of sentence segmentation is to find the boundaries of the sentences.
- The *Topic Segmentation* task aims to divide a stream of text or speech into topically homogeneous blocks. Given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics change.
- The *Name Tagging* task aims to mark the names (persons, locations, and organizations) in a text.

For relatively simpler tasks, such as *Turkish Text Deasciifier*, *Word Segmentation*, and *Vowel Restoration*, only lexical information is enough, but in order to obtain

better performance in more complex tasks, such as *Sentence Segmentation*, *Topic Segmentation*, and *Name Tagging*, we not only use lexical information, but also exploit morphological, and contextual information. For sentence segmentation, we have modeled the final inflectional groups of the words and combined it with the lexical model, and decreased the error rate to 4.34%. For name tagging, in addition to the lexical and morphological models, we have also employed contextual and tag models, and reached an F-measure of 91.56%. For topic segmentation, stems of the words (nouns) have been found to be more effective than using the surface forms of the words and we have achieved 10.90% segmentation error rate on our test set.

Keywords: Information Extraction, Statistical Natural Language Processing, Turkish, Named Entity Extraction, Topic Segmentation, Sentence Segmentation, Vowel Restoration, Word Segmentation, Text Deasciification.

ÖZET

TÜRKÇE İÇİN İSTATİSTİKSEL BİR BİLGİ ÇIKARIM SİSTEMİ

Gökhan Tür
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Doç. Dr. Kemal Oflazer
Ağustos, 2000

Bu tezde, istatistiksel dil işleme yöntemleri kullanarak Türkçe metinlerden bilgi çıkarımı üzerine yapılan bir dizi çalışmanın sonuçları sunulmaktadır. Sözcüksel (lexical) ve biçimbirimsel (morphological) bilgiler kullanan istatistiksel yöntemler aşağıdaki problemlerde başarıyla uygulanmıştır:

- *Türkçe Metin Düzeltme* sistemi, ASCII karakter kümesinde olmayan Türkçe karakterlerin ASCII karşılıklarıyla (ör: "ı" yerine "i") yazıldıkları metinleri düzeltme amacını taşır.
- *Sözcüklere Ayırma* sistemi, içinde boşluk ya da noktalama işaretleri olmayan bir dizi karakter verildiğinde, bunları sözcüklerine ayırmaya çalışır.
- *Ünlüleri Yerine Koyma* sistemi, ünlü karakterleri olmayan bir metin verildiğinde bunları tekrar yerine koymayı amaçlar.
- *Cümlelere Ayırma* sistemi, bir dizi sözcük verildiğinde bunları sözdizimsel cümlelere bölmeyi amaçlar.
- *Konulara Ayırma* sistemi, bir metinde konuların değiştiği yerleri bulmayı amaçlar.
- *İsim İşaretleme* sistemi, bir metindeki özel isimleri (insan, yer, ve kurum isimleri) işaretlemeyi amaçlar.

Türkçe Metin Düzeltme, *Sözcüklere Ayırma*, ve *Ünlüleri Yerine Koyma* gibi görece basit sistemler için sözcüksel bilginin yeterli olduğu görüldü. Ancak *Cümlelere Ayırma*, *Konulara Ayırma*, ve *İsim İşaretleme* gibi daha karmaşık

problemler için, ek olarak biçimbirimsel ve çevresel (contextual) bilgi de kullanıldı. Cümlelere ayırma problemi için, sözcüklerin son çekim eki grubunu (inflectional group) istatistiksel modelleyip sözbirimsel modelle birleştirerek hata oranını 4.34%'e düşürmeyi başardık. İsim işaretleme sisteminde, sözbirimsel ve biçimbirimsel modellerin yanı sıra, çevresel ve işaret (tag) modellerini de kullandık ve 91.56% oranında doğruluğa ulaştık. Konulara ayırma problemi için ise, sözcüklerin köklerini kullanmak, asıl hallerini kullanmaktan daha iyi sonuçlar verdi, ve hata oranı 10.90% oldu.

Anahtar sözcükler: Bilgi Çıkarımı, İstatistiksel Doğal Dil İşleme, Türkçe, İsim İşaretleme, Konulara Ayırma, Cümlelere Ayırma, Ünlüleri Yerine Koyma, Sözcüklere Ayırma, Türkçe Metin Düzeltme.

Acknowledgment

I would like to express my gratitude to my supervisor Kemal Oflazer for his guidance, suggestions, and invaluable encouragement over the last 6 years at Bilkent University.

This work was begun while I was visiting Speech Technology and Research Laboratory, SRI International, as an international fellow between July 1998 and July 1999. It was a pleasure working with Andreas Stolcke and Elizabeth Shriberg. I learned a lot from them. I also would like to thank Fred Jelinek of Johns Hopkins University, Center for Language and Speech Processing, who suggested me and my wife, Dilek, to this lab.

I would like to thank the members of Johns Hopkins University, Eric Brill, Fred Jelinek, and David Yarowsky, for introducing me to statistical language processing in general while I was visiting Computer Science Department of this university between September 1997 and June 1998.

I would like to thank A. Enis Çetin, Bilge Say, Özgür Ulusoy, and İlyas Çiçekli for reading and commenting on this thesis.

I am grateful to my family and my friends for their infinite moral support and help throughout my life.

Finally, I would like to thank my wife, Dilek for her endless effort and support during this study, while she was doing her own thesis. This thesis would not be possible without her determination.

To my parents and my wife Dilek

Contents

1 Introduction **1**

1.1 Information Extraction 1

1.2 Approaches to Language and Speech Processing 4

1.3 Motivation 7

1.4 Thesis Layout 10

2 Statistical Information Theory **11**

2.1 Statistical Language Modeling 13

2.1.1 Smoothing 19

2.2 Hidden Markov Models 21

3 Turkish **25**

3.1 Morphology 26

3.2 Inflectional Groups (IGs) 27

3.3 Morphological Disambiguation 28

3.4 Potential Problems 29

4 Simple Statistical Applications 30

4.1 Introduction 30

4.2 Turkish Text Deasciifier 30

4.2.1 Introduction 30

4.2.2 Previous Work 31

4.2.3 Approach 32

4.2.4 Experiments and Results 33

4.2.5 Error Analysis 34

4.3 Word Segmentation 36

4.3.1 Introduction 36

4.3.2 Approach 36

4.3.3 Experiments and Results 37

4.3.4 Error Analysis 38

4.4 Vowel Restoration 38

4.4.1 Introduction 38

4.4.2 Approach 39

4.4.3 Experiments and Results 39

4.4.4 Error Analysis 40

4.5 Conclusions 41

5 Sentence Segmentation 42

5.1 Introduction 42

5.2 Previous Work 43

5.3 Approach 46

5.3.1 Word-based Model 47

5.3.2 Morphological Model 47

5.3.3 Model Combination 48

5.4 Experiments and Results 48

5.4.1 Training and Test Data 49

5.4.2 Evaluation Metrics 49

5.4.3 Results 49

5.4.4 Error Analysis 51

5.5 Conclusion 52

6 Topic Segmentation **53**

6.1 Introduction 53

6.2 Previous Work 56

6.2.1 Approaches based on word usage 56

6.2.2 Approaches based on discourse and combined cues 59

6.3 The Approach 61

6.3.1 Word-based Modeling 63

6.3.2 Stem-based Modeling 64

6.3.3 Noun-based Modeling 67
6.4 Experiments and Results 70
6.4.1 Training Data 70
6.4.2 Test Data 70
6.4.3 Evaluation metrics 71
6.4.4 Segmentation Results 72
6.4.5 Error Analysis 74
6.4.6 Results Compared to Topic Segmentation of English . . . 74
6.4.7 False Alarm vs. Miss Rates 75
6.4.8 The Effect of Chopping 75
6.5 Conclusion 77

7 Name Tagging **79**

7.1 Introduction 79
7.2 Task Definition 81
7.2.1 Organizations 83
7.2.2 Locations 83
7.2.3 Persons 84
7.3 Previous Work 84
7.3.1 Rule-based Approaches 85
7.3.2 Machine Learning Approaches 88

7.3.3 Hybrid Approaches 93
7.4 Motivation 95
7.5 Approach 97
7.5.1 Lexical Model 98
7.5.2 Contextual Model 100
7.5.3 Morphological Model 101
7.6 Tag Model 103
7.7 Model Combination 104
7.8 Experiments and Results 107
7.8.1 Training and Test Data 107
7.8.2 Evaluation Metrics 107
7.8.3 Results 110
7.8.4 Error Analysis 111
7.8.5 Effect of the Case and Punctuation Information 112
7.8.6 Results Compared to Name Tagging of English 113
7.9 Conclusion 114

8 Conclusion **116**

A Turkish Morphological Features **121**

B Turkish Stopword List **123**

List of Figures

1.1 An example of an example broadcast news word transcript, whose named entities are marked. *ENAMEX* tags names, such as person, location, and organization, *TIMEX* tags time expressions, such as date or time 2

1.2 An example of a topic boundary in a Turkish newspaper. 3

2.1 Schematic diagram of a general communication system. 12

2.2 An HMM used to tag a text with 3 parts-of-speech, Noun, Verb, and Adjective (Adj). 24

4.1 The HMM for the input word "ışık". 32

5.1 Examples of sentence boundaries in a football news article. <**S**> denotes a sentence boundary. 42

5.2 The conceptual figure of the HMM used by SRI for sentence segmentation. YB denotes that there is a sentence boundary, NB denotes that there is no sentence boundary, $WORD$ denotes the words of the text. 45

6.1 An example of a topic boundary in a broadcast news word transcript. 54

6.2 An example of a topic boundary in a Turkish newspaper. 55

6.3 Structure of the basic HMM developed by Dragon for the TDT Pilot Project. The labels on the arrows indicate the transition probabilities. TSP represents the topic switch penalty. 59

6.4 Structure of the final HMM with fictitious boundary states used for combining language and prosodic models. In the figure, states B1, B2, ..., B100 represent the presence of a topic boundary, whereas states N1, N2, ..., N100 represent topic-internal sentence boundaries. TSP is the topic switch penalty. 62

6.5 False alarm versus miss probabilities for automatic topic segmentation of news for both development (Dev) and test (Test) sets. . 76

7.1 An example of an example broadcast news word transcript, whose named entities are marked. 80

7.2 The conceptual structure of the basic HMM used by BBN for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, `per` denotes person, `loc` denotes location, `org` denotes organization, and `else` denotes that it does not belong to any of these categories. 89

7.3 An example of an example Turkish news article, whose named entities are marked. 96

7.4 The conceptual structure of the basic HMM for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, `yes` denotes the name boundary, `no` denotes that there is no name boundary, `mid` denotes that it is in the middle of a name, `per` denotes person, `loc` denotes location, `org` denotes organization, and `else` denotes that it does not belong to any of these categories. 99

7.5 Combining lexical, contextual, morphological, and tag models for tagging Turkish text. 108

List of Tables

1.1 Comparison of the number of unique word forms in English and Turkish, in large text corpora. 8

1.2 The frequency table for the root word *gol* (goal) observed in a sport news corpus. 9

2.1 The character-based entropy and perplexity values for English and Turkish. All results for English have been obtained using a trigram language model. 18

2.2 The word-based entropy and perplexity values for English and Turkish using a trigram language model. 18

2.3 Good-Turing estimates for bigrams from 22 million AP bigrams. . 20

2.4 The state observation likelihoods of the HMM states for each word. Note that the columns add up to 1. 23

2.5 The state transition probabilities of the HMM states. Note that the rows add up to 1. 24

3.1 Numbers of analyses and IGs in Turkish 28

4.1 Results for Turkish text deasciifier. 34

4.2 Distribution of the errors for Turkish text deasciifier. 35

4.3 Results for Turkish word segmentor. 38

4.4 Results for Turkish vowel restoration system. 40

5.1 Results for sentence segmentation on Broadcast News (boundary recognition error rates). Values are error rates (in percent). 45

5.2 The effect of the word-based language model. 47

5.3 The effect of the word-based language model. 48

5.4 Results for Turkish sentence segmentation using word-based, morphological language models, and their combinations. *LM* denotes the word-based model, and *MM* denotes the morphological model. *Baseline* denotes the performance, when we put a sentence boundary after every finite verb. 50

5.5 Confusion matrix for Turkish sentence segmentor. "Sent" denotes a sentence boundary, whereas, "Else" denotes a non-sentence boundary. 51

6.1 The most frequent words in one of the clusters, containing mostly football news articles. *Loc* denotes locative case, *Acc* denotes accusative case. 64

6.2 The frequency table for the root word *gol* (goal) in the cluster mentioned in Table 6.1. 66

6.3 The most frequent stems in a cluster, containing mostly football news articles. 68

6.4 The most frequent nouns in a cluster, containing mostly football news articles. 69

6.5 Summary of error rates with different language models. A "chance" classifier that labels all potential boundaries as non-topic would achieve 0.3 weighted segmentation cost. "Random" indicates that the articles are shuffled. 73

6.6 The unigram probabilities of the words in the example sentence. Note that, the word *son* (last) is a stopword, hence gets 0 probability. 74

6.7 Word-based segmentation error rates for English and Turkish corpora. 75

6.8 Word-based segmentation error rates using word-based models, when we use fixed length sentences, or when we use the sentence boundaries marked by the automatic sentence segmentor, or when they are given (True Boundaries). 77

7.1 MUC-7 Name Tagging Scores for English. 85

7.2 The effect of the boundary flag on the performance of the tagger. 100

7.3 The use of the contextual model for unknown words. 100

7.4 The use of the morphological model. 103

7.5 The use of the tag model. 104

7.6 An example output of the MUC scorer. 110

7.7 Accuracy of the name tagging task using lexical, contextual, morphological, and tag models. 110

7.8 Detailed name tagging results. 111

7.9 Accuracy of the name tagging task using lexical, contextual, and tag models without case and punctuation information. 113

7.10 Comparison of the Turkish and English name tagging results using only lexical and contextual models. 114

8.1 Summary of the tasks, and the information sources other than the lexical information used in this thesis. 117

Chapter 1

Introduction

This thesis presents the results of a study on information extraction from unrestricted Turkish text using statistical language processing methods. The thesis first describes the notion of information extraction, and itemize the main components of an information extraction system. Then it discusses the two main approaches to language and speech processing: statistical and knowledge based approaches. Subsequently, it presents the properties of Turkish in order to point the major problems in building a statistical information extraction system for Turkish.

1.1 Information Extraction

Information extraction (IE) is the task of extracting particular types of entities, relations, or events from natural language text or speech. The notion of what constitutes information extraction has been heavily influenced by the *Message Understanding Conferences* (MUCs) [MUC, 1995; MUC, 1998; Grishman, 1998; Grishman and Sundheim, 1996]. This conference has been extended also to handle other languages, such as Spanish, Japanese, and Chinese in the Multilingual Entity Task (MET) conferences. A relatively new conference also related to information extraction is the *Topic Detection and Tracking Conference* (TDTs)

...The other very big story of the <TIMEX TYPE="DATE">today</ENAMEX> is in <ENAMEX TYPE="LOCATION">Washington</ENAMEX> where the <ENAMEX TYPE="ORGANIZATION">White House</ENAMEX> administration has already been badly shaken up by the possibility that president <ENAMEX TYPE="PERSON">Clinton</ENAMEX> and one of his advisors <ENAMEX TYPE="PERSON">Vernon Jordan</ENAMEX> obstructed justice. ...

Figure 1.1: An example of an example broadcast news word transcript, whose named entities are marked. *ENAMEX* tags names, such as person, location, and organization, *TIMEX* tags time expressions, such as date or time

which refers to automatic techniques for finding topically related material in streams of data (e.g., newswire and broadcast news) [Wayne, 1998].

The following are some of the common IE tasks:

- The *Named Entity Extraction* task covers marking names (persons, locations, and organizations), and certain structured expressions (money, percent, date and time). In this task, finding only names is called *name tagging.* An example text, whose named entities are marked, is given in Figure 1.1.

- The *Coreference* task covers noun phrases (common and proper) and personal pronouns that are "identical" in their reference; it requires production of tags for coreferring strings from *equivalence* classes. For instance, in the above example, the word *his* at the last sentence is referring to the president.

- The *Template Element* task covers organizations, persons, and artifacts, which are captured in the form of template *objects* consisting of a predefined set of attributes. For instance, a template element for Clinton may look like the following:

```
<ENTITY-0592-3> :=
ENT_NAME: "Bill Clinton"
ENT_TYPE: PERSON
ENT_DESCRIPTOR: "president"
```

...Ardından Bursa panik yaptı , Nihat ustalık dolu bir vuruşla beraberliği sağladı. Beşiktaş ucuz kurtuldu. <**TOPIC_CHANGE**> Enerji Zirvesi'nin onur konuğu ABD eski Başkanı George Bush , dünyanın refahı için global projelerde birleşmenin şart olduğuna dikkat çekti ...

Figure 1.2: An example of a topic boundary in a Turkish newspaper.

`ENT_CATEGORY: PER_CIV`

- The *Template Relation* task requires identifying relationships between template elements. Example relationships are `PRODUCT_OF`, `EMPLOYEE_OF`, `LOCATION_OF`, etc.

- The *Scenario Template* task requires identifying instances of a task-specific event and identifying event attributes, including entities that fill some role in the event; the overall information content is captured via interlinked *objects*.

- The *Topic Segmentation* task deals with the problem of automatically dividing a stream of text into topically homogeneous blocks. That is, given a sequence of words, the aim is to find the boundaries where topics change. A topic (or a story) is defined to be a seminal event or activity, along with all directly related events and activities. An example topic change is demostrated in Figure 1.2

- The *Topic Detection* task tries to associate stories to topics.

- The *Topic Tracking* task tries to detect the stories related to a given topic.

- The *Sentence Segmentation* task deals with automatically dividing a stream of text or speech into grammatical sentences. Given a sequence of (written or spoken) words without any punctuation or case information, the aim of sentence segmentation is to find the boundaries of the sentences.

In this thesis, we only deal with *name tagging*, *sentence segmentation*, and *topic segmentation* tasks, in the context of unrestricted Turkish text, and how

statistical approaches can be used for them. But beforehand, in order to give the reader the flavor of the statistical methods in speech and language processing, we will also present statistical approaches for some simple tasks, such as *word segmentation*, which deals with automatically dividing a stream of characters into legitimate words, *deasciifier*, which deals with converting Turkish text written using ASCII characters to Latin-5 characters, and a *vowel restoration* system which tries to restore the vowels of an input stream, whose vowels are deleted.

1.2 Approaches to Language and Speech Processing

Until recently, natural language processing technology has used symbolic approaches, while speech recognition technology has traditionally used statistical approaches [Price, 1996; Charniak, 1993; Church and Mercer, 1993; Young and Bloothooft, 1997]. For many years, the use of statistics in language processing was not so popular. The following famous quote of Chomsky [1969] represents the sentiment of that period:

> "It must be recognized that the notion of a *probability of a sentence* is an entirely useless one, under any interpretation of this term."

On the other hand, speech recognition community was working on stochastic methods [Jelinek *et al.*, 1975; Jelinek, 1998; Bahl *et al.*, 1983], inspired by the early studies on information theory [Shannon, 1948; Jelinek, 1968]. In 80s, Jelinek, then the director of the IBM speech group, replied to Chomsky during a workshop:[1]

> "Anytime a linguist leaves the group the recognition rate goes up."

[1] Although this quote was not written down until 90s [Palmer and Finin, 1990], I am sure of it, because Jelinek repeated the same utterance when I was taking his course during my visit to Johns Hopkins University, Department of Computer Science!

Integration of these technologies with a *balancing act* [Klavans and Resnik, 1996] is a promising research area. Beginning from late 1980s, in the context of projects funded by DARPA, these two cultures began to merge, and currently there are highly sophisticated systems that contain both statistical and linguistic approaches.

In early 90s, the Association of Computational Linguistics published a special issue of the Journal of Computational Linguistics (CL) on using large corpora [CL93, 1993]. The call-for-papers for this special issue expresses this change in the minds:

> "The increasing availability of machine-readable corpora has suggested new methods for studies in a variety of areas such as lexical knowledge acquisition, grammar construction, and machine translation. Though common in speech community, the use of statistical and probabilistic methods to discover and organize data is relatively new to the field at large. ... Given the growing interest in corpus studies, it seems timely to devote an issue of CL to this topic."

In this issue, Church claims that probabilistic models provide a theoretical abstraction of language, very much like Chomsky's competence model [Church and Mercer, 1993]. They are designed to capture the more important aspects of language and ignore the less important ones. What counts as important depends on the application. For example, if you consider the part-of-speech tagging task,[2] in Brown corpus, the word "bird" appears as a noun in 25 times out of 25, and "see" appears as a verb in 771 times out of 772. However, it is possible to see "bird" as a verb, or "see" as a noun in dictionaries. In these cases, traditional methods have tended to ignore the lexical preferences, which are very important in such a task. Attempts to eliminate unwanted tags using only syntactic information is sometimes not very successful. For example, the trivial sentence "I see a bird" can be tagged as "I/Noun see/Noun a/Noun bird/Noun", as in "city/Noun school/Noun committee/Noun meeting/Noun".

[2]Part-of-speech tagging task tries to determine the correct syntactic category (i.e. part-of-speech tag, such as *verb*, or *noun*) of the words

Note that, most of the resistance to probabilistic techniques has two main reasons:

1. The misconception about using statistics: While building a system using statistical methods, the linguistic knowledge about that specific task is said to be ignored. This is not the case; instead, this knowledge is used to guide the modeling process and to enable improved generalization with respect to unseen data [Young and Bloothooft, 1997].

2. The vagueness of n-grams during 60s: In order to obtain a useful language model, it is necessary to have enough training data. Because of this reason, Shannon's n-gram approximation was long left unstudied and therefore Chomsky introduced an alternative with complementary strengths and weaknesses. For example, his approximation is much more appropriate for modeling long-distance dependencies [Church and Mercer, 1993].

We will not attempt to explain completely these two schools of computational linguistics in this thesis. Instead, we would like to summarize the advantages and disadvantages of the two approaches, noting that, in general, one's weakness is the strength of the other's.

Statistical models have the following advantages [Price, 1996; Young and Bloothooft, 1997; Appelt and Israel, 1999]:

- They can be trained automatically (provided there is enough data), which facilitates their porting to new domains and uses.

- The probabilities can directly be used as scores, thus, they can provide a systematic and convenient mechanism for combining multiple knowledge sources.

- Weak and vague dependencies can be modeled easily. For example, a very rarely seen word sequence can still get some probability.

- Domain portability is relatively straightforward.

- System expertise is not required for customization.

On the other hand, they have the following disadvantages:

- Generally best performing systems are obtained using knowledge-base approaches.
- Training data may not exist. This is especially important for lesser studied tasks and languages.
- Standard n-gram language models have certain weaknesses, such as data sparseness and insufficiency in modeling long distance relationships, although there are some number of studies in order to overcome this problem [Rosenfeld, 1994; Chelba, 2000].
- Changes to specifications may require reannotation of large quantities of training data.

1.3 Motivation

In contrast to languages like English, for which there is a very small number of possible word forms with a given root word, languages like Turkish or Finnish with very productive agglutinative morphology where it is possible to produce thousands of forms (or even millions [Hankamer, 1989]) for a given root word, pose a challenging problem for statistical language processing.

In Turkish, using the surface forms of the words results in data sparseness in the training data. Table 1.1 shows the size of the vocabulary obtained by a recent study conducted by Hakkani-Tür [2000] on about 10 million word corpora of Turkish and English, collected from online newspapers.

In order to demonstrate the effect of this data sparseness, consider Table 1.2. This table presents a list of different formations of the stem word *gol* (goal),

Language	Vocabulary Size
English	97,734
Turkish	474,957

Table 1.1: Comparison of the number of unique word forms in English and Turkish, in large text corpora.

observed in a sport news corpus.[3] Thus, it is a necessity for Turkish, more than English, to analyze the words morphologically in order to build models for various tasks.

Hakkani-Tür [2000] proposes methods for statistical language modeling of Turkish. She uses the inflectional groups (IGs) in the morphological analyses of the words in order to build a language model for Turkish, and proves the effectiveness of this method in the statistical morphological disambiguation of Turkish. An IG is a sequence of inflectional morphemes, separated by derivation boundaries. We follow her idea of using IGs and the morphological analyses of the words depending on the task. The method of using the morphological information in IE tasks forms a motivation of this thesis.

On the other hand, statistical methods have been largely ignored for processing Turkish. Mainly due to the agglutinative nature of Turkish words and the structure of Turkish sentences, the construction of a language model for Turkish can not be directly adapted from English. It is necessary to incorporate some other techniques. In this sense, this work is a preliminary step in the application of corpus-based statistical methods to Turkish text processing.

Another motivation for this study is that, there is no known system for Turkish dealing with any of the information extraction tasks described above, though there are several information retrieval and language processing systems for Turkish [Hakkani-Tür, 2000; Tür, 1996; Oflazer, 1993; Hakkani *et al.*, 1998; Oflazer, 1999, among others]. In our view, regardless of the method and technologies used, developing such a system for the first time for Turkish is as important

[3]The morphological features used in this word are given in Appendix A.

Word	Freq	Morphological Analysis
gol	1222	goal+Noun+A3sg+Pnon+Nom
golü	350	goal+Noun+A3sg+Pnon+Acc or
		goal+Noun+A3sg+P3sg+Nom
gole	150	goal+Noun+A3sg+Pnon+Dat
golle	138	goal+Noun+A3sg+Pnon+Ins
goller	126	goal+Noun+A3pl+Pnon+Nom
golde	85	goal+Noun+A3sg+Pnon+Loc
golün	75	goal+Noun+A3sg+Pnon+Gen or
		goal+Noun+A3sg+P2sg+Nom
golünü	63	goal+Noun+A3sg+P3sg+Acc or
		goal+Noun+A3sg+P2sg+Acc
golüyle	62	goal+Noun+A3sg+P3sg+Ins
golcü	59	goal+Noun+A3sg+Pnon+Nom^DB+Adj+Agt
golleri	48	goal+Noun+A3pl+P3sg+Nom or
		goal+Noun+A3pl+Pnon+Acc or
		goal+Noun+A3pl+P3pl+Nom or
		goal+Noun+A3sg+P3pl+Nom
golden	45	goal+Noun+A3sg+Pnon+Abl
gollerle	40	goal+Noun+A3pl+Pnon+Ins
gollük	37	goal+Noun+A3sg+Pnon+Nom^DB+Adj+FitFor
gollü	26	goal+Noun+A3sg+Pnon+Nom^DB+Adj+With
golüne	24	goal+Noun+A3sg+P3sg+Dat or
		goal+Noun+A3sg+P2sg+Dat
golleriyle	20	goal+Noun+A3pl+P3sg+Ins or
		goal+Noun+A3pl+P3pl+Ins or
		goal+Noun+A3sg+P3pl+Ins
golsüz	18	goal+Noun+A3sg+Pnon+Nom^DB+Adj+Without
golcüsü	18	goal+Noun+A3sg+Pnon+Nom^DB+Noun+Agt+A3sg+P3sg+Nom
golünde	16	goal+Noun+A3sg+P3sg+Loc or
		goal+Noun+A3sg+P2sg+Loc
gollerde	15	goal+Noun+A3pl+Pnon+Loc
goldeki	15	goal+Noun+A3sg+Pnon+Loc^DB+Det
gollerin	12	goal+Noun+A3pl+Pnon+Gen or
		goal+Noun+A3pl+P2sg+Nom
golünden	10	goal+Noun+A3sg+P3sg+Abl or
		goal+Noun+A3sg+P2sg+Abl
gollerini	9	goal+Noun+A3pl+P3sg+Acc or
		goal+Noun+A3pl+P2sg+Acc or
		goal+Noun+A3pl+P3pl+Acc or
		goal+Noun+A3sg+P3pl+Acc
gollere	8	goal+Noun+A3pl+Pnon+Dat

Table 1.2: The frequency table for the root word *gol* (goal) observed in a sport news corpus.

as developing an information retrieval, a speech recognition, or a machine translation system for Turkish.

1.4 Thesis Layout

The organization of this thesis is as follows: Chapter 2 explains Shannon's information theory with examples from both Turkish and English; Chapter 3 deals with the characteristics of Turkish; Chapter 4 presents some simple tasks in order to give the reader the flavor of the statistical methods in speech and language processing; Chapter 5 explains our work on statistical sentence segmentation of words without punctuation and case information using lexical and morphological information; Chapter 6 presents a topic segmentation system using only the nouns of the sentences; Chapter 7 presents a Turkish name tagging system, using lexical, contextual, and morphological information. Finally, we conclude with Chapter 8.

Chapter 2

Statistical Information Theory

In this thesis, we will follow the information theory of Shannon [Shannon, 1948]. In this theory, Shannon defines information as a purely quantitative measure of communicative exchanges. A communication is defined as a system of five parts as depicted in Figure 2.1:

1. An *information source* which produces the message(s) to be communicated to the receiving terminal,
2. A *transmitter* which operates on the message in some way to produce a signal suitable for transmission over the channel,
3. The *channel*, which is the medium used to transmit the signal the signal from transmitter to receiver,
4. The *receiver*, which ordinarily performs the inverse operation of that done by the transmitter, reconstructing the message from the signal, and
5. The *destination*, which is the person or thing, for whom the message is intended.

A communication system can be classified into three main categories:

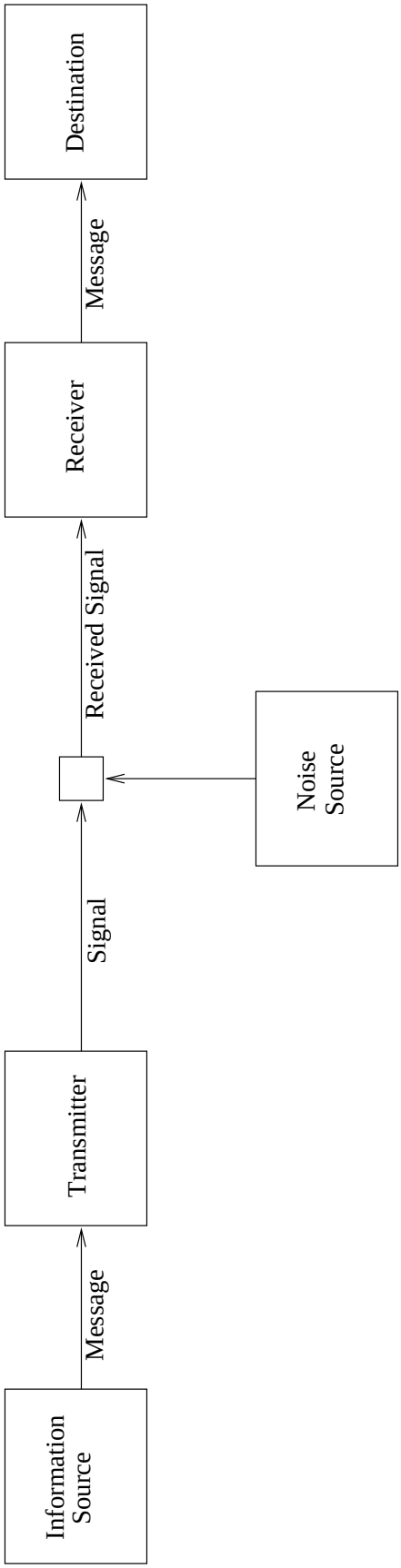

Figure 2.1: Schematic diagram of a general communication system.

1. *Discrete systems*: Both the message and the signal are a sequence of discrete symbols. For example, sequence of letters forming a text.

2. *Continuous systems*: The message and the signal are both treated as continuous functions. For example, radio or television broadcasts.

3. *Mixed systems*: Both discrete and continuous variables appear. For example Pulse-Code Modulation (PCM) transmission of speech.

Since we are dealing with language processing, we will only consider discrete case. We can think of a discrete source as generating the message, symbol by symbol. It will choose successive symbols according to certain probabilities depending on preceding choices as well as the particular symbols in question. A physical system, or a mathematical model of a system which produces such a sequence of symbols, governed by a set of probabilities, is known as a *stochastic process*. Thus, we may consider a discrete source to be represented by a stochastic process. Such stochastic processes are known mathematically as discrete Markov processes and have been extensively studied in the literature. Markov models are the class of probabilistic models, that assume that we can predict the probability of some future model without looking at too far into the past. An n^{th} order Markov model looks $n-1$ words into the past. In computational linguistic terms, this called as an $(n-1)^{th}$ order statistical language model. In this thesis, we employ only statistical language models and hidden Markov models (HMM). In this section, we will briefly describe these concepts. Detailed explanations can be found in numerous related books [Cover and Thomas, 1991; Manning and Schütze, 1999; Jelinek, 1998; Charniak, 1993; Jurafsky and Martin, 2000].

2.1 Statistical Language Modeling

Statistical language models root back to Shannon's early work on information theory [Shannon, 1948]. Their aim is basically to predict the probability of the next word, given the previous words, $P(w_i|w_1, ..., w_{i-1})$.

Guessing the next word correctly has interestingly many applications in language and speech processing. For example, in speech recognition, it is very important in choosing among various candidate words.

Statistical modeling of word sequences is called language modeling. Since we cannot possibly consider each history, $w_1, w_2, ..., w_n$ separately, as this would imply very large sample space, we group the histories according to their last $n-1$ words to obtain an n-gram language model. This gives us,

$$P(w_i|w_1, ..., w_{i-1}) \approx P(w_i|w_{i-n+1}, ..., w_{i-1})$$

For example, in a trigram language model, this probability is obtained using the previous two words:

$$P(w_i|w_1, ..., w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1})$$

It is easy to obtain these n-gram probabilities from a corpus by counting the number of occurrences of the n-grams, according to the maximum likelihood estimation [Jurafsky and Martin, 2000]:

$$P(w_i|w_{i-n+1}, ..., w_{i-1}) \approx \frac{C(w_{i-n+1}, ..., w_i)}{C(w_{i-n+1}, ..., w_{i-1})}$$

where $C(w_i, ..., w_j)$ is the number of occurrences of the word sequence $w_i, ..., w_j$ in the corpus.

For example, for a trigram language model, we can rewrite the above formula as follows:

$$P(w_i|w_{i-2}, w_{i-1}) \approx \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})}$$

Given an n-gram language model, it is straightforward to compute the probability of a sentence including $w_1, w_2, ..., w_m$ using the formula:

$$P(w_1, w_2, ..., w_m) = \prod_{k=1}^{m} P(w_i | w_{i-n+1}, ..., w_{i-1})$$

where the words with negative indices can be ignored.

For example, for a trigram language model, this formula becomes:

$$P(w_1, w_2, ..., w_m) = P(w_1) \times P(w_2|w_1) \times \prod_{k=3}^{m} P(w_k | w_{k-2}, w_{k-1})$$

Although it is possible to use language models for modeling any sequence, we would like to give some word-based or character-based language model examples, as we are dealing with natural language processing. We have built a language model for Turkish, using about 18 million words of Milliyet newspaper web resources covering a period from January 1997 to September 1998. The following is the most probable word sequence according to this language model:

"Çünkü , bu işten mümkün kumarhaneler bunu işine karıştırmamalı. Manisa Savcılığı tarafından yürüttüğünü ve ancak penaltıdan fark olduğunu belirterek , " Kayırmacı haber bülteninde oranı yüzde , kendilerine 13. Bizim ait politikacılarına Genel Sekreteri Orhan Dk."

Jurafsky has trained a language model for English using a book of Shakespeare [Jurafsky and Martin, 2000]. According to this model, a corresponding example for English has given as follows:

"Sweet prince, Falstaff shall die. Harry of Monmouth's grave. This should forbid it should be branded, if renown made it empty. What isn't that cried?"

In order to see the effect of the training data, consider the similar experiment using a trigram language model trained on Wall Street Journal news articles:

"They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates stores as Mexico and Brazil on market conditions"

A corresponding experiment for character-based language models has also

been performed by Shannon [1948] using a trigram language model. We also repeated this experiment using a trigram language model built using the same training data.

- English:
  "in no ist lat whey cratict froure birs grocid pondenome of demonstures of the reptagin is regoactiona of cre"
- Turkish:
  "Tümerdinin bir ya . Vekişmazırlarının çalı"

In order to show the effect of the order of a language model, consider the same experiment using 6-gram character-based language model:

"Simitis'le bir yazan Dk . 65 milyondan yazık ki, mermisine kadar alıp , Beşiktaşlar ile ABD'ye eminin kişisel tam uygulaması yapamaması birbirleşerek şöyle birkaş kez daha sonra dediğini gösteriyordu"

Entropy and *perplexity* are the most common metrics used to evaluate n-gram models. Entropy is a measure of information, and is invaluable in language and speech processing. It can be used as a metric for how much information there is in a particular model, for how well a language model matches a given language. Entropy is defined as follows [Shannon, 1948]:

$$H(X) = -\sum_{x \epsilon X} p(x) log_2 p(x)$$

where the random variable X ranges over whatever we are predicting (words, letters, parts of speech, etc.).

Although it is possible to take logarithm in any base, in order to measure entropy in terms of bits, it is generally convenient to use base 2. In this case, entropy can be interpreted as the minimum number of bits it would take to encode a certain piece of information.

The entropy of a language, $H(L)$, is defined to be the limit of the per-symbol entropy as the length of message gets very large. Then the above formula becomes:

$$H_p(L) = -\lim_{n\to\infty} \frac{1}{n} \sum_{w_1,...,w_n \in L} p(w_1,...,w_m) log_2 p(w_1,...,w_n)$$

Since we do not know the actual probability distribution p, and use a model m instead, we use cross entropy. Cross entropy can be proven to be greater than or equal to the actual entropy.

$$H_{p,m}(L) = -\lim_{n\to\infty} \frac{1}{n} \sum_{w_1,...,w_n \in L} p(w_1,...,w_n) log_2 p_m(w_1,...,w_n)$$

If the language L is stationary and ergodic, this limit can be stated as:

$$H_m(L) = -\lim_{n\to\infty} \frac{1}{n} log_2 p_m(w_1,...,w_n)$$

A stochastic process is said to be stationary if the probabilities it assigns to a sequence are invariant with respect to the shifts in the time index. Markov models, hence n-grams are stationary. But natural languages are not stationary. Thus our statistical models only give an approximation to the correct distributions and entropies of natural language. A language is said to be ergodic if any sample of the language, if made long enough, is such a perfect sample[1].

Using the formulas above, it is possible to compute the cross entropy of a corpus, given a language model m. Shannon [1948] reported a per-letter entropy of 1.3 bits for English. In a later study, using much larger training data (583 million words) to create a trigram language model, and much larger test corpus (1 million words), this number has been shown to be 1.75 bits [Brown *et al.*, 1992]. In our experiments, we have found a per-letter entropy of 2.02 bits for Turkish, using a 6-gram language model trained by using 18 million words with 120 million characters, on a test corpus of 76,524 characters.

[1]See [Cover and Thomas, 1991; Manning and Schütze, 1999; Jelinek, 1998; Charniak, 1993; Jurafsky and Martin, 2000] for details and proof of this theorem

	Corpus Size	Entropy	Perplexity
Turkish (trigram)	18M words	3.14	8.84
Turkish (6-gram)	18M words	2.02	4.06
English (Shannon)	<1M words	1.30	2.46
English (Brown)	583M words	1.75	3.36

Table 2.1: The character-based entropy and perplexity values for English and Turkish. All results for English have been obtained using a trigram language model.

	Corpus Size	Entropy	Perplexity
Turkish	18M words	10.21	1188
English (Shannon)	<1M words	7.15	142
English (Brown)	583M words	6.77	109
English (Hakkani-Tür)	10M words	6.77	109

Table 2.2: The word-based entropy and perplexity values for English and Turkish using a trigram language model.

The value 2^H is called the *perplexity*. Perplexity can intuitively be thought of as the weighted average number of choices a random variable has to make. For example choosing among 8 equally likely choices, where entropy is 3 bits, the perplexity would be $2^3 = 8$. In other words, a perplexity of k means that you are as surprised on average as you would have been if you had to guess between k equiprobable choices at each step.

Using the same training and test data, Brown has reported a perplexity of 109 for English [Brown *et al.*, 1992]. The word level perplexity of Turkish, on the other hand is significantly larger[2] [Hakkani-Tür, 2000]. In Tables 2.1 and 2.2, we summarize the entropy and perplexity results for Turkish and English for both character and word-based models. These results are important as they shed light on problems in statistical modeling of Turkish.

An intuitive way of obtaining a word entropy from a character entropy is to multiply the character entropy with the average word length, For example, in

[2]In Hakkani-Tür's work, the perplexity of English has been found to be same as Brown.

English this length is 5.5 characters. This is how the perplexity of Shannon's model is obtained from the character entropy. Although this computation does not hold for other results exactly, there is a correlation between the word and character entropies. This must be the reason for the higher entropy of Turkish character-based language model.

2.1.1 Smoothing

Even when we use a trigram model, with a vocabulary size of 20,000, there are 8×10^{12} probabilities to estimate. Because of this sparseness problem, it is necessary to employ one of the available methods for smoothing these probabilities.

In this section, we are going to discuss two methods for smoothing. The first one is a discounting method, called "Good-Turing", the other one relies on the n-gram hierarchy, called "Back-off". Although there are other methods of smoothing, we will not consider them, since in all of the tasks discussed in this thesis, we are going to use the Good-Turing discounting, combined with back-off, except in the task of topic segmentation. It is one of the few language and speech processing tasks, in which smoothing decrases the performance. We are going to discuss this issue, in Chapter 6. The reason for using Good-Turing with back-off is that, these methods are widely accepted to perform best on most of the tasks [Church and Gale, 1991].

Good-Turing Smoothing

The Good-Turing smoothing algorithm was first described by Good [1953], who credits Turing with the original idea: Re-estimate the amount of probability mass to assign to n-grams with zero or low counts by looking at the number of n-grams with higher counts.

$$P_{GT} = \frac{c^*}{N}$$

c	N_c	c^*	P_{GT}
0	74,671,100,000	0.0000270	1.23×10^{-12}
1	2,018,046	0.446	2.03×10^{-8}
2	449,721	1.26	5.73×10^{-8}
3	188,933	2.24	1.02×10^{-7}
4	105,668	3.24	1.47×10^{-7}
5	68,379	4.22	1.9×10^{-7}

Table 2.3: Good-Turing estimates for bigrams from 22 million AP bigrams.

where N is the training data size, and

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

where N_c is the number of n-grams occuring c times. N_0 is defined as the number of all unseen n-grams. For example, if we deal with bigrams, N_0 will be equal to the square of the vocabulary size, minus all the bigrams we have seen.

Table 2.3 demonstrates the use of Good-Turing smoothing, from 22 million AP bigrams. In this table, first column indicates the c values, and the second column indicates the frequencies of the frequencies. For example, according to this table, the number of bigrams occuring 5 times is 68,379. The last column indicates the probabilities, which are given to the corresponding bigrams. An unseen bigram gets a probability of 1.23×10^{-12}, whereas a bigram occured 5 times gets a probability of 1.9×10^{-7} according to the formulas.

In practice, this discounted estimate c^* is not used for all counts of c. Large counts (where $c > k$ for some threshold k) are assumed to be reliable. For example, $k = 5$ is said to be a good threshold to select.

Back-off Smoothing

Another method for smoothing is the *back-off modeling* proposed by Katz [1997]. The estimate for the n-gram is allowed to back off through progressively shorter

histories. If the n-gram did not appear at all or appeared k times or less in the training data, then we use an estimate from a shorter n-gram. More formally, for $n = 3$, and $k = 0$:

$$P_{bo}(w_i|w_{i-2}, w_{i-1}) = \begin{cases} P(w_i|w_{i-2}, w_{i-1}) & if\ C(w_{i-2}, w_{i-1}, w_i) > 0 \\ \alpha_{w_{i-1},w_i} P(w_i|w_{i-1}) & else\ if\ C(w_{i-2}, w_{i-1}, w_i) = 0 \\ & and\ C(w_{i-1}, w_i) > 0 \\ \alpha_{w_i} P(w_i) & otherwise \end{cases}$$

where P_{bo} is the back-off probability, $C(w_i, ..., w_j)$ is the number of occurrences of the word sequence $w_i, ..., w_j$ in the corpus, function d is used for the amount discounted, and α is the normalizing factor, obtained using a formula, which guarantees that the sum of all probabilies add up to 1.

2.2 Hidden Markov Models

A hidden Markov model (HMM) is a probabilistic model, modeling a sequence of events [Rabiner and Juang, 1986]. For example, for part-of-speech tagging task, the part-of-speech tag of a word is a random event with a probability that can be estimated from an annotated training data.

In an HMM, there is an underlying finite state machine (whose states are not directly observable, hence hidden) that changes state with each input element. So constructing an HMM recognizer depends on two things:

- constructing a good hidden state model (For example, for the part-of-speech tagging task, it is straightforward to use one state for each part-of-speech.) and,
- examining enough training data to accurately estimate the probabilities of the various state transitions given sequences of words.

More formally, an HMM is specified by a four-tuple (S, O, P, Q), where:

- S is the set of states s_i with a unique starting state s_0,
- O is the output alphabet $o_0, ..., o_n$,
- P is a probability distribution of transitions, $p(s_i|s_j)$, between states, also called as state transition probabilities, and
- Q is the output probability distribution, $q(o_i|s_j)$, also called as state observation likelihoods.

Then, the probability of observing an HMM output string $o_1, o_2, ..., o_k$ is given by:

$$P(o_1, ..., o_k) = \sum_{s_1,...,s_k} \prod_{i=1}^{k} p(s_i|s_{i-1})q(o_i|s_i)$$

It is possible to use the probabilities obtained from the language models as state transition probabilities in an HMM. For example, for part-of-speech tagging task, the transition from the state of *Noun* to the state of *Verb* is nothing but $P(Verb|Noun)$.

What we are trying to do is to find the tag sequence, T, which maximizes the probability $P(T|W)$, for the input stream W, i.e.

$$\underset{T}{\operatorname{argmax}}\, P(T|W) \tag{2.1}$$

According to the Bayes' rule we get the formula:

$$P(T|W) = \frac{P(W|T)P(T)}{P(W)}$$

Note that $P(W)$ is given, hence constant, thus Equation 2.1 equals to:

$$\underset{T}{\operatorname{argmax}}\, P(W|T)P(T) \tag{2.2}$$

	w_1	w_2	w_3	w_4
Noun	0.2	0.4	0.1	0.9
Verb	0.3	0.4	0.4	0.05
Adjective	0.5	0.2	0.5	0.05

Table 2.4: The state observation likelihoods of the HMM states for each word. Note that the columns add up to 1.

In an HMM the state observation likelihoods determine the probability of observing the input string W, given the state sequence T, i.e. $P(W|T)$. Similarly, the state transition probabilities give the probability of following the state sequence T, i.e. $P(T)$.

When we use an HMM, the probability of observing an HMM output string is thus nothing, but $P(T|W)P(W)$. Then it is enough to compute the maximum likelihood path through the hidden state model for the input word sequence, W, (which is the output string of the HMM), thus marking spans of input correspond to marking states. The search algorithm usually used to find such a path is called the *Viterbi* algorithm [Viterbi, 1967]. This dynamic programming algorithm is well explained in the literature on speech recognition [Jelinek, 1998; Jurafsky and Martin, 2000; Manning and Schütze, 1999; Charniak, 1993]. The maximum likelihood path gives the state sequence that maximizes the Equation 2.2, hence the Equation 2.1.

Let's consider a simplified part-of-speech tagging task, in which we have only 3 tags, say, *Noun*, *Verb*, and *Adjective*. It is possible to use a 3 state HMM, where each state outputs words of that part-of-speech tag as shown in Figure 2.2. Assume that we would like to tag our input "w_1 w_2 w_3 w_4". The state observation likelihoods for our 4 words, for these 3 states are given in Table 2.4. For example, $q(w_1|Noun) = 0.2$. These likelihoods may also be obtained from the training data. If 20% of the time w_1 was tagged as Noun, then its likelihood can be set to 0.2.

Now, we can also define the state transition probabilities using a matrix. Table 2.5 shows the example bigram probabilities. For example, $p(Verb|Noun) = 0.3$.

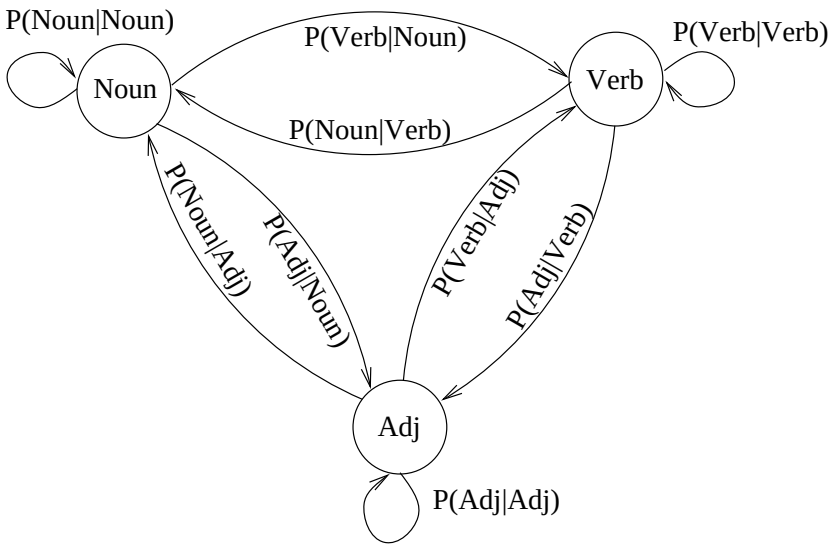

Figure 2.2: An HMM used to tag a text with 3 parts-of-speech, Noun, Verb, and Adjective (Adj).

	Noun	Verb	Adjective
Noun	0.5	0.3	0.2
Verb	0.4	0.3	0.3
Adjective	0.4	0.2	0.4

Table 2.5: The state transition probabilities of the HMM states. Note that the rows add up to 1.

These probabilities may be obtained from the language model.

For such an HMM, the most probable path goes from the states "Adjective Noun Verb Noun" in that order. In fact, this gives us the most probable part of speech sequence for this example.

In this thesis, in order to build and use a language model, and decode the most probable output in an HMM with the Viterbi algorithm, we used the publicly available SRILM toolkit, developed by Andreas Stolcke [Stolcke, 1999].

Chapter 3

Turkish

Turkic languages constitute the sixth most widely spoken language in the world, and spread over a large geographical area in Europe and Asia. It is spoken in Turkish, Azeri, Türkmen, Tartar, Uzbek, the Baskurti, Nogay, Kyrgyz, Kazakh, Yakuti, Cuvas and other dialects. Turkish belongs to the Altaic branch of the Ural-Altaic family of languages, and has the following major properties:

- Agglutinative morphology,
- Free constituent order in a sentence,
- Head-final structure.

This chapter will focus on only the morphological aspects of Turkish, since this is the single most important characteristic for the tasks presented in this thesis. Note that, for more complex IE tasks, such as scenario element, the last two items would be critical, since then it would be necessary to (light) parse a sentence.

3.1 Morphology

Turkish is an agglutinative language, in which a sequence of inflectional and derivational morphemes can be added to a word [Oflazer, 1993]. The number of word forms one can derive from a root form may be in the millions [Hankamer, 1989]. For instance, the derived modifier *sağlamlaştırdığımızdaki* (Literally, "(the thing existing) at the time we caused (something) to become strong") would be morphologically decomposed as:

sağlam+laş+tır+dı+ğı+mız+da+ki

and morphologically analyzed as:[1]

```
sağlam+Adj^DB
+Verb+Become^DB
+Verb+Caus+Pos^DB
+Adj+PastPart+P1sg^DB
+Noun+Zero+A3sg+Pnon+Loc^DB
+Adj
```

In order to have an idea of the productivity of Turkish morphology, also see Table 1.2 for a list of different formations of the stem word *gol* (goal), observed in a sport news corpus.

A Turkish morphological analyzer has been developed by Oflazer [1993] using the two-level finite-state transducer technology developed by Xerox [Karttunen, 1993]. In this thesis, we use this system to obtain the morphological analyses of the words.

[1] The morphological features used in this word are given in Appendix A.

3.2 Inflectional Groups (IGs)

A Turkish word can be represented as a sequence of *inflectional groups* (IGs) as described by Oflazer [1999]. An IG is a sequence of inflectional morphemes, separated by derivation boundaries (^DB). For example, the above word, *sağlamlaştırdığımızdaki*, would be represented with the following 6 IGs:

1. sağlam+Adj
2. Verb+Become
3. Verb+Caus+Pos
4. Adj+PastPart+P1sg
5. Noun+Zero+A3sg+Pnon+Loc
6. Adj

We have used the final IGs of the words in name tagging and topic segmentation tasks, for the following two reasons:

- The final IG determines the final category, hence its function of a word. For example, our example word is unlikely to be a sentence final word, since its final category is adjective. Recall that Turkish is a head-final language, i.e. sentences generally end with a finite verb.
- The use of the final IG instead of the whole morphological analysis solves the problem of data sparseness. While there may be theoretically infinitely many such word forms in Turkish, the number of possible final IGs is limited. Table 3.1 presents the number of IGs observed in a corpus of 1 million words [Hakkani-Tür, 2000].

	Possible	Observed
Full Analyses (No roots)	∞	10,531
Inflectional Groups	9,129	2,194

Table 3.1: Numbers of analyses and IGs in Turkish

3.3 Morphological Disambiguation

This extensive use of suffixes in Turkish causes morphological parsing of words to be rather complicated, and results in ambiguous lexical interpretations in many cases. For example, the word "çocukları" is 4-way ambiguous:

1. child+Noun+A3pl+P3sg+Nom (his children)
2. child+Noun+A3sg+P3pl+Nom (their child)
3. child+Noun+A3pl+P3pl+Nom (their children)
4. child+Noun+A3pl+Pnon+Acc (children) (Acc)

The disambiguation of Turkish is a well studied area in Turkish text processing. Kuruöz and Oflazer [1994; 1994], then Tür and Oflazer [1997; 1996; 1996] have used rule-based methods, Hakkani-Tür and Oflazer [2000; 2000] have used a statistical approach for this problem. In these studies, the accuracy of the morphological disambiguation is found to be about 95% regardless of the method used.

In thesis, whenever we needed to use morphological information, we either left the ambiguity as is (such as in topic segmentation task), or used the statistical morphological disambiguation system developed by Hakkani-Tür [2000] (such as in sentence segmentation and name tagging).

3.4 Potential Problems

In this section, we will list some potential problems of building a statistical system for Turkish.

- The most important problem for using statistical methods for Turkish is the data sparseness, because of the agglutinative nature of the language. As given in Tables 2.1 and 2.2, the perplexity of Turkish is much higher than that of English. In order to build a successful statistical model, it is necessary to incorporate morphological information for non-trivial tasks.
- Being a lesser-studied language especially for information extraction related tasks, there are no annotated corpora for training and testing purposes.
- Being a lesser-studied language using statistical methods, we have little sources of reference in order to compare our results, too.

Chapter 4

Simple Statistical Applications

4.1 Introduction

In this chapter, we will present some simple tasks, and how statistical approaches can be used for them. In order to give the reader the flavor of the statistical methods in speech and language processing, we have tried the following three simple tasks:

- Turkish text deasciifier,
- Vowel restoration, and
- Word segmentation

4.2 Turkish Text Deasciifier

4.2.1 Introduction

There is quite an amount of on-line Turkish text which is typed using an ASCII character set where non-ASCII Turkish characters are typed using their nearest

ASCII equivalent, e.g., *ü* is entered as *u* , etc. Otherwise these texts are just fine with almost no other errors and hence are quite useful for NLP research. In this task, our aim is to convert such texts into their correct forms, i.e. to deasciify them. For instance the text

"Bu kosullarda Turkiye'nin benimseyecegi akilci hareket tarzi elindeki kozlari heba etmeden AB'ye tam uyelik icin yesil isik yakilan ikinci grup ulkelerin en basinda yer almayi hedeflemektedir."

would be converted to

"Bu koşullarda Türkiye'nin benimseyeceği akılcı hareket tarzı elindeki kozları heba etmeden AB'ye tam üyelik için yeşil ışık yakılan ikinci grup ülkelerin en başında yer almayı hedeflemektedir."

Although seems like a trivial task, there are lots of potential problems, and it is very hard to convert all of the characters correctly due to the ambiguity. We will analyze the errors in Section 4.2.5.

4.2.2 Previous Work

Mutlu Uysal developed such as a system as a senior project, using the Error-tolerant Finite State Recognition Algorithm of Oflazer [1996], with additional heuristics and statistics to "fix the text." This algorithm works as follows: First, it takes the word to be deasciified, then generates possible candidates according to the ambiguous characters. For example, for the word "isik", it generates the following 8 possible candidates:

1. "ışık"
2. "ışik"
3. "ısık"
4. "ısik"

5. "işık"
6. "işik"
7. "ısık"
8. "isik"

The system then chooses the correct form, by searching in the finite state transducer for Turkish developed for morphological analysis by Oflazer [1993]. Additional heuristics and statistics were used to resolve ambiguous cases like "su" and "şu". This system was reported to have an error rate of 2%.

4.2.3 Approach

Our approach was very simple in this task. Using our 18 million word corpus, we built a character-based language model using the SRILM toolkit [Stolcke, 1999]. Then we built an HMM in which states denote the characters, and the transition probabilities were obtained from the language model. The order of the language model was a parameter of our system. The state observation likelihhods were set to 1, so that effectively we used only the language model in Viterbi decoding. We did not use any other information source in this task.

While decoding, we built an HMM, in which all ambiguous characters ('i','o','u','c','g',and 's') are represented by two states, whereas other characters are represented by a single state. The corresponding HMM for the input "ışık" is given in Figure 4.1.

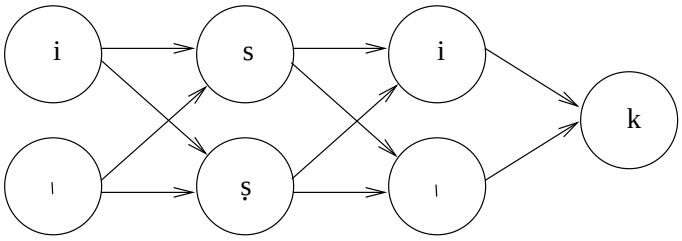

Figure 4.1: The HMM for the input word "ışık".

4.2.4 Experiments and Results

Training and Test Data

In order to build the language model, we used the web resources of Milliyet newspaper articles, covering the period from January 1, 1997 through September 12, 1998, containing about 18 million words, 119,057,159 characters. We evaluated the performance of the deasciifier, using a test data of 8,511 words, 91,468 characters. In this set, there are 16,770 ambiguous characters in 5,864 words that needed to be resolved.

Evaluation Metrics

In order to evaluate the performance of our system, we used the error rate, which can be computed as follows:

$$Error\ Rate = \frac{Number\ of\ Wrong\ Characters}{Number\ of\ Ambiguous\ Characters}$$

That means, for the input word "ışık", if we decoded it as "işık", then the error rate would be $\frac{1}{3} = 33.33\%$.

Results

Table 4.1 shows our performance using a character-based language model. The baseline performance for this task was 33.45% error, obtained by choosing the most probable choice among the candidates. For example, it favored for "i" instead of 'ı', etc. This was exactly the same result as using the unigram language model for this task. When we increased the order of the language model, the performance increased drastically, and finally converged to around 1% error rate. It was interesting to see a huge decrease in the error rate when we used 4-gram instead of 3-gram language model. This indicates that a context of 2 characters is not enough to decide on the third character for Turkish. This may be due to the

Model	Error Rate (%)
1-gram	33.55
2-gram	28.96
3-gram	11.84
4-gram	3.14
5-gram (pruned)	1.58
5-gram	1.12
6-gram	1.04
Uysal and Oflazer	1.95

Table 4.1: Results for Turkish text deasciifier.

fact that the average length of the stems of the words is more than 3 characters. We also tried a pruned version of the 5-gram model, in order to speed up this decoding. In that case, our system became about 20 times faster by losing only 0.5% in error rate.

In order to compare our performance with the previous deasciifier of Uysal and Oflazer described in Section 4.2.2, we ran that system on our test data. Their performance appeared to be 1% less than ours. However, that system was designed to work in monocase data, hence it was a little misleading to compare both systems. Because in some cases, case information becomes critical, as in "kanı" vs. the proper name "Kani".

4.2.5 Error Analysis

When we look at the errors made our best performing system, we end up with the error distribution showed in Table 4.2. It is interesting to see that our system favors for "i" instead of "ı".

Since our letter-based model is checking for local context, it would be useful to incorporate this system with morphological analyzer, and check whether it is a legitimate word. In that case, errors made in long words can be eliminated. Here are some errorful words, which can be corrected easily: *çıkmayin*. It is also a solution to interpolate our letter-based model with a word-based model, and

Character	Number of Errors	Number of Occurances	Error Rate (%)
i	25	4922	0.50
İ	1	71	1.40
ı	33	2470	1.33
I	0	4	0
ü	32	1137	0.28
u	26	1539	0.17
U	0	8	0
Ü	0	13	0
ö	3	410	0.73
o	9	1319	0.68
Ö	0	31	0
O	0	35	0
Ş	1	26	0.04
S	2	107	1.86
s	16	1628	0.98
ş	16	809	0.20
ç	2	477	0.41
Ç	1	27	3.70
c	6	508	1.18
C	0	25	0
ğ	2	526	0.38
Ğ	0	1	0
G	0	88	0
g	0	589	0
Total	175	16770	1.04

Table 4.2: Distribution of the errors for Turkish text deasciifier.

check for illegitimate words and word sequences. Such an interpolation may also correct errors of the type *Türk koyu* vs. *Türk köyü*.

4.3 Word Segmentation

4.3.1 Introduction

In this task, our aim is to detect word boundaries, given we have a sequence of characters. For example, the text

"bukoşullardatürkiyeninbenimseyeceğiakılcıharekettarzıelindekikozlarıhebaetme denabyetamüyelikiçinyeşilışıkyakılanikincigrupülkelerinenbaşındayeralmayıhedef lemektedir."

would be converted to

"bu koşullarda türkiyenin benimseyeceği akılcı hareket tarzı elindeki kozları heba etmeden abye tam üyelik için yeşil ışık yakılan ikinci grup ülkelerin en başında yer almayı hedeflemektedir."

This task has an application in speech recognition. Given that we have recognized phones, we can convert them to a sequence of letters. Then all we need to do is segment them into words. Recognition of phones is a much simpler task than recognition of words, especially for agglutinative languages for Turkish. This is why, we have ignored case and punctuation information for this task.

4.3.2 Approach

Our approach is very similar to the one we have used in the deasciifier task. We built a character language model using our 18 million word training data. The only difference is that, between each character we put a boundary flag, in order to mark whether there is a word boundary or not. For example, the the input "bizim ev" is converted to "Y b N i N z N i N m Y e N v Y", where Y denotes a word boundary, and "N denotes otherwise. We used this language model in order to determine the state transition probabilities in an HMM, in which states denote either a character or a boundary flag (Y or N). Between two characters,

the output sequence must pass through a boundary flag character. Using the Viterbi algorithm, we ended up with the most probable segmentation of an input stream.

4.3.3 Experiments and Results

Training and Test Data

In order to build the language model, we used the web resources of Milliyet newspaper articles, covering the period from January 1, 1997 through September 12, 1998, containing about 18 million words, 11,9057,159 characters. We evaluated the performance of the word segmentor, using a test data of 1294 words, 8898 characters excluding spaces and punctuation from a Turkish history text.

Evaluation Metrics

In order to evaluate the performance of our system, we used the error rate, which can be computed as follows:

$$Error\ Rate = \frac{Number\ of\ False\ Alarms + Number\ of\ Misses}{Number\ of\ Characters}$$

Results

Table 4.4 shows our performance in the word segmentation task. The baseline performance for this task is 14.5% error, obtained by labeling all locations as non-boundaries (the most frequent class). By looking at a window of 9 tokens (characters or boundaries), we have achieved less than 0.5% error rate.

Model	Error Rate (%)
1-gram	14.50
2-gram	12.96
3-gram	8.34
4-gram	6.27
5-gram	4.08
6-gram	1.97
7-gram	1.10
8-gram	0.69
9-gram	0.49

Table 4.3: Results for Turkish word segmentor.

4.3.4 Error Analysis

Most of the errors made on the short words, which can also be used as suffixes, such as "de", "da", "ki", "mi", etc. The other category prone to errors is the compound words, especially verbs, such as "çekegeldiği".

4.4 Vowel Restoration

4.4.1 Introduction

In this task, our aim is to restore the vowels of an input stream, whose vowels are deleted. For example, the text

"b kşllrd trkynn bnmsycğ klc hrkt trz lndk kzlr hb tmdn by tm ylk çn yşl şk ykln knc grp lklrn n bşnd yr lmy hdflmktdr"

would be converted to

"bu koşullarda türkiyenin benimseyeceği akılcı hareket tarzı elindeki kozları heba etmeden abye tam üyelik için yeşil ışık yakılan ikinci grup ülkelerin en başında yer almayı hedeflemektedir."

A possible application of this would be vocalization of text in Ottoman archives. These texts were written using Arabic alphabet and most vowels were omitted. For example, in order to write "mektep", they use "mim-kef-te-be" in arabic alphabet, similarly they use "kef-ye-dal-he-be-ye-lam-re-sin-nun" in order to write "gidebilirsin".

4.4.2 Approach

Similar to the deasciifier, we have built a character language model, then we built an HMM, in which after each consonant, it is possible to choose from 8 different vowels plus a special character for no-vowel (NV) case. For example, "evde" can be recognized as "e v NV d e". Note that this representation fails to capture words in which there are two consecutive vowels, like "saat". We ignored such cases, because our performance decreased when we include these cases in our HMM. The transition probabilities were obtained from the language model, and the state observation likelihoods were set to 1, as in the previous two tasks.

4.4.3 Experiments and Results

Training and Test Data

In order to build the language model, we used the web resources of Milliyet newspaper articles, covering the period from January 1, 1997 through September 12, 1998, containing about 18 million words, 11,9057,159 characters. We evaluated the performance of the vowel restoration system, using a test data of 1294 words, 8898 characters from a Turkish history text.

Evaluation Metrics

In order to evaluate the performance of our system, we used the error rate, which can be computed as follows:

Model	Error Rate (%)
1-gram	59.75
2-gram	50.50
3-gram	41.02
4-gram	32.31
5-gram	16.66
6-gram	12.35
7-gram	9.98
8-gram	9.35
9-gram	9.42

Table 4.4: Results for Turkish vowel restoration system.

$$Error\ Rate = \frac{Number\ of\ Wrong\ Characters}{Number\ of\ Candidates}$$

Candidates indicate locations in which it is possible to use a vowel, i.e. before and after each consonant. For example, for the input word "vd" (for "evde"), if we decoded it as "vade", then the error rate would be $\frac{2}{3} = 66.67\%$.

Results

Table 4.4 shows our performance in the word segmentation task. The baseline performance for this task is 59.75% error, obtained by doing nothing, i.e. marking all candidates as non-vowels (the most frequent class). By looking at a window of 8 tokens (characters or boundaries), we have achieved 9.35% error rate.

4.4.4 Error Analysis

This task is very prone to errors. Even the performance of the human is expected to be very poor in this task. There are lots of cases in which it is possible to interpret the words in many ways, such as restoring "vd" as "evde", "veda", "vade", "vaadi", "avda", "ovdu", "avdı", "evdi", "övdü", "vadi", or "ivedi".

Since our model is character based, instead of word-based, it is also possible to produce illegitimate words such as "totraktör" for "ttrk" (for "Atatürk"). Thus, similar to the deasciifier, it would be useful to incorporate this system with morphological analyzer, and check whether it is a legitimate word. It is also a solution to interpolate our letter-based model with a word-based model, and check for illegitimate words and word sequences.

4.5 Conclusions

In this chapter we tried to demonstrate how statistical approaches can be used for language processing with three simple tasks. We have tried only language modeling in order to develop these systems, and used no other information. Using the SRILM toolkit, it was very fast to develop the models, build the HMM, and run Viterbi algorithm. We are very pleased to obtain very satisfactory results, for all these three tasks. These example systems are important as the reader will get prepared for forthcoming more complex tasks of sentence segmentation, topic segmentation, and name tagging.

Chapter 5

Sentence Segmentation

5.1 Introduction

Sentence segmentation is the task of automatically dividing a stream of text or speech into grammatical sentences. Given a sequence of (written or spoken) words, the aim of sentence segmentation is to find the boundaries of the sentences. Figure 5.1 gives examples of sentence boundaries from a football news article.

Note that the sentences inside quotes are not considered as separate sentences, if they occur inside another sentence. That means sentences are not recursive

<S> toshack eğer taşları yerinden oynatmazsa yani çılgınlık yapmazsa beşiktaş'ı şampiyonluğun adayları arasında göstermiştik <S> ama adamın yapısı belli <S> toshack ikinci yarıda ertuğrul'u oyundan alıp yerine nihat'ı sokarken oyun düzeninde değişikliğe gitti ve oktay'ı ohen'in yanına çekti <S> nitekim bu değişiklik biraz olsun kartal'ı hem baskıdan kurtardı hem de rakip savunmada çoğalmayı sağladı <S> ne olduysa oktay'ın inanılmaz golünden sonra oldu <S> ardından bursa panik yaptı <S> nihat ustalık dolu bir vuruşla beraberliği sağladı<S>

Figure 5.1: Examples of sentence boundaries in a football news article. <S> denotes a sentence boundary.

structures. The following example contains only one sentence:

Ahmet "Bugün okula gidiyorum. Çok sevinçliyim" dedi.
(Literally, "Ahmet said that he was going to school today, (and) he was very happy.")

whereas there are 2 different sentences in the following example:

Ahmet dedi ki: <**S**> "Bugün okula gidiyorum. Çok sevinçliyim."
(Literally, "Ahmet said: "I am going to school today. I am very happy")

Sentence segmentation is a preliminary step towards speech understanding. Many natural language and speech processing tasks, such as parsing the sentence, finding topic changes, aligning multilingual text, require their input to be divided into sentences. Once the sentence boundaries have been detected, then further syntactic and/or semantic analysis can be performed on these sentences. Furthermore, speech recognizer output lacks the usual textual cues to these entities (such as headers, paragraphs, sentence punctuation, and capitalization).

5.2 Previous Work

In this work, we limited our task to finding sentence boundaries, when there was no punctuation or case information, assuming our input was a speech recognizer output.

This task was studied in SRI International, STAR Lab for English [Stolcke *et al.*, 1998; Stolcke *et al.*, 1999; Shriberg *et al.*, 2000; Hakkani-Tür *et al.*, 1999]. They tried to combine the lexical model with the prosodic model. The lexical information was modeled using an n-gram language model, trained from 130 million annotated words. They built an HMM, as depicted in Figure 5.2, in which states either denote whether there was a boundary or not between two words, or denote the words in the text. Transition probabilities were obtained from the language model. Besides this lexical model, they built a separate prosodic model using a decision tree which classifies the word boundaries as sentence boundary

or non-sentence boundary. This prosodic model was trained using the prosodic features obtained from 700,000 words of broadcast news transcripts. In order to combine these two models, they proposed two methods:

1. *Posterior probability interpolation*: Both the decision tree used for modeling prosody and the language model estimate posterior probabilities for each boundary type T. These probabilities are then interpolated as given in the following formula:

$$P(T|W,F) \approx \lambda P_{\mathrm{LM}}(T|W) + (1-\lambda) P_{\mathrm{DT}}(T|F,W)$$

   where W denotes the word sequence, and F denotes the prosodic information, modeled using prosodic features, such as pause duration, pitch, etc. inbetween the words. λ is a parameter optimized on held-out data to optimize the overall model performance, LM denotes the language model, and DT denotes the decision tree, i.e. prosodic model.

2. *Integrated hidden Markov modeling*: In fact, the HMM used for lexical modeling can be extended to "emit" both words and prosodic observations. Using Bayes' rule:

$$\underset{T}{\operatorname{argmax}}\, P(T|W,F) = argmax_T P_{\mathrm{LM}}(T|W) \times P(F|T,W)$$

   Posteriors obtained from the prosodic model, $P_{\mathrm{DT}}(T_i|F_i,W)$, can be used as likelihoods, $P(F|T,W)$, when the decision tree uses downsampled training data, so that $P(T|W) = \mathbf{yes}) = P(T|W) = \mathbf{no}) = \frac{1}{2}$.

$$P(F|T,W) = \frac{P(F|W) P_{\mathrm{DT}}(T|F,W)}{P(T|W)}$$

since $P(F|W)$ is a constant for all choices of T.

They found out that prosody was a very valuable source in detecting sentence boundaries from speech recognizer output. Table 5.1 shows the results on both transcribed (true) and recognized words, for sentence segmentation models for

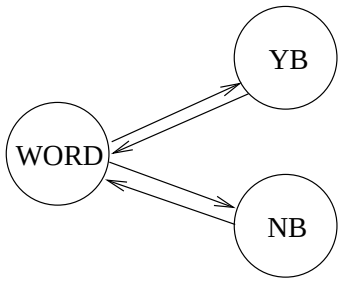

Figure 5.2: The conceptual figure of the HMM used by SRI for sentence segmentation. YB denotes that there is a sentence boundary, NB denotes that there is no sentence boundary, $WORD$ denotes the words of the text.

Model	Transcribed Words (%)	Recognized Words (%)
LM only (130M words)	4.1	11.8
Prosody only (700K words)	3.6	10.9
Interpolated	3.5	10.8
Combined HMM	3.3	11.7
Chance	6.2	13.3
Lower bound	0.0	7.9

Table 5.1: Results for sentence segmentation on Broadcast News (boundary recognition error rates). Values are error rates (in percent).

Broadcast News corpus. The baseline (or "chance") performance for true words in this task was 6.2% error, obtained by labeling all locations as nonboundaries (the most frequent class). For recognized words, it was considerably higher; this was due to the non-zero lower bound resulting if one accounts for locations in which the 1-best hypothesis boundaries do not coincide with those of the reference alignment. "Lower bound" gives the lowest segmentation error rate possible given the word boundary mismatches due to recognition errors.

Even with the punctuation marks, this task is not easy. We will not give the details of these systems, as with the punctuation marks, it is possible to achieve more than 99% accuracy. Reynar and Rathnaparkhi [1997] have presented a maximum entropy approach for this task. Palmer and Hearst [1997] have integrated neural networks with decision trees in order to detect sentence boundaries. Other researchers have used regular grammars, or some simple rules

to detect boundaries even without mentioning this task.

5.3 Approach

Like all other tasks described in this thesis, a statistical model was used for this task. We tried to group the words into contiguous stretches belonging to one sentence, i.e., the word boundaries were classified into "sentence boundaries" and "non-sentence boundaries". Sentence segmentation task was thus reduced to a boundary classification problem. We will use B to denote the string of binary boundary classifications, and W to denote the word sequence. Our approach aimed to find the segmentation B with highest probability given the information in W.

$$\underset{B}{\operatorname{argmax}}\, P(B|W)$$

We formed our training data, so that each word was followed by a boundary flag which denotes whether there was a sentence boundary or not. For example the portion from the Figure 5.1:

... çoğalmayı sağladı <**S**> ne olduysa oktay'ın ...

was converted to

... NB çoğalmayı NB sağladı YB ne NB olduysa NB oktay'ın NB ...

where YB denotes that there is a boundary, and NB denotes otherwise. As our input lacked punctuation, case, or other acoustic and prosodic information, we had no source of information other than words. We have made use of the surface forms and morphological analyses of the words.

Output Sequence	Probability
geldi NB çünkü	0.00028166
geldi YB çünkü	0.00614714

Table 5.2: The effect of the word-based language model.

5.3.1 Word-based Model

We built a language model using only surface forms of the words similar to SRI system. This model also enabled us to gauge our baseline performance.

The language model was formed from the training data, as described above. In order to see the effect of this model, consider the portion "... geldi çünkü ...". Table 5.2 shows the probabilities of the possible taggings for this text piece. As seen, the one with the boundary has about 30 times more probable than the other.

5.3.2 Morphological Model

In addition to the surface forms of the words, we used the morphological analyses of the words, which hold valuable information for this task, and alleviate the data sparseness problem we would encounter in building the language model.

While forming the morphological model, we used the final inflectional groups[1] of the morphological analyses of the words instead of the surface forms.

In order to build the morphological model, we used a preprocessing module, developed by Hakkani-Tür [2000], which tokenizes the training data, analyzes the tokens using the morphological analyzer developed by Oflazer [1993], groups the collocations, removes some obviously improbable morphological parses in order to reduce the morphological ambiguity, and finally gives the most probable morphological analyses.

[1]See Chapter 3 (IGs) for a detailed explanation of inflectional groups.

Output Sequence	Probability
Verb+Pos+Past+A3sg NB	0.24849
Verb+Pos+Past+A3sg YB	0.751505

Table 5.3: The effect of the word-based language model.

Table 5.3 shows the probabilities of the possible taggings after the word "geldi" (came) according to the morphological model. The word "geldi" is morphologically analyzed as "*Verb+Pos+Past+A3sg*"[2]. As seen, it is about 3 times more probable of marking a sentence boundary after a final verb.

5.3.3 Model Combination

We prefered the posterior probability interpolation method of SRI, in order to combine these two information sources. Instead of the prosodic model, we now have the morphological model.

$$P(T|W, M(W)) \approx \lambda P_{\mathrm{WM}}(T|W) + (1-\lambda) P_{\mathrm{MM}}(T|M(W))$$

where WM denotes the word-based model, MM denotes the morphological model, T denotes the boundary type, W denotes the word sequence, $M(W)$ denotes the morphological analyses of W, λ is a parameter optimized on held-out data to optimize the overall model performance.

5.4 Experiments and Results

In order to evaluate the word-based and morphological model, and their combined performance, we carried out experiments described in this section. We used SRILM toolkit for language modeling and decoding [Stolcke, 1999]. We first

[2]See Appendix A for the meanings of these features in the morphological analyses.

describe our training and test data, then give results obtained with the word-based, morphological language models and their combinations.

5.4.1 Training and Test Data

The word-based model was trained using the web resources of Milliyet newspaper articles, covering the period from January 1, 1997 through September 12, 1998, containing about 18 million words, 50,674 sentences. In order to see the effect of the training size, we also used a small subset of 1 million words from this corpus for training. Test data contains 14738 words, 931 sentences from the same newspaper.

5.4.2 Evaluation Metrics

In order to evaluate our system, we followed the evaluation criteria used in SRI sytem in order to obtain comparable results. According to this metric, each word boundary is marked as sentence or non-sentence boundary, and we align the output with manually annotated test data, and finally compute the error rate with the following formula:

$$Error\ Rate = \frac{Number\ of\ False\ Alarms + Number\ of\ Misses}{Number\ of\ Words}$$

5.4.3 Results

Table 5.4 shows our performance using word-based and morphological models, and their combinations. The chance performance for this task is 6.65% error, obtained by labeling all locations as non-sentence boundaries (the most frequent class). The baseline performance was obtained by marking sentence boundaries to the locations, where the preceding words' morphologically analyses contain Verb tag in their final IGs, except the cases where the word was analyzed as conditional verb as in "gelirse" (literally "if s/he comes"), or as imperative verb as in "gel"

Model	Error Rate (%)
Chance	8.65
Baseline	5.85
LM only (1M words)	5.98
LM only (18M words)	4.82
MM only (1M words)	4.90
MM only (18M words)	4.90
LM (18M) + MM (1M)	4.59
LM (18M) + MM (18M)	4.34

Table 5.4: Results for Turkish sentence segmentation using word-based, morphological language models, and their combinations. *LM* denotes the word-based model, and *MM* denotes the morphological model. *Baseline* denotes the performance, when we put a sentence boundary after every finite verb.

(literally "Come!"). We ignored imperative verbs, because such analyses occured most of the time, if the word was mis-analyzed, or if it was in a quotation.

Results show that the morphological model alone performs better than a word-based language model, unless the language model was trained on a much larger data set. This is a typical result of the data sparseness we have encountered while training the word-based model. Training with 1 million words performed even worse than the baseline. An interesting result is that, the morphological model performed similarly when trained with 1 million and 18 million words, although this similarity disappeared interestingly when combined with the word-based model. Also it is worthwhile to note that it is possible to get close performances with the morphological model trained with 1 million words, instead of a word-based model trained with 18 times more data. Most importantly, statistically significant error reductions of 21% and 25% over the baseline were achieved by combining the word-based model trained with 18 million words with the morphological model trained with 1 million and 18 million words consequtively.

We would like to give the false alarm and miss error distribution in Table 5.5 for our best performing system, the combination of word-based and morphological models, both trained using 18 million words of data. This confusion matrix indicates that out of 467 errors the system has made, 203 of them are false

		Detected	
		Sent	Else
True	Sent	668	264
	Else	203	9633

Table 5.5: Confusion matrix for Turkish sentence segmentor. "Sent" denotes a sentence boundary, whereas, "Else" denotes a non-sentence boundary.

alarms, and 264 of them are missed boundaries.

These results are very similar to the results obtained for English as given in Section 5.2, encouraging us for better results when prosodic information is also incorporated.

5.4.4 Error Analysis

When we analyzed our errors, we saw 3 major categories of errors:

1. The system sometimes made errors while deciding to end the sentence after the words, which could be used as final verb, or derived adjective, such as:

   "purşasb düzenlediği basın toplantısında afganistan sınırında bu ay sonu düzenlenecek <**S**> zülfikar tatbikatı için kara kuvvetlerinin bölgeye bin asker sevkedeceğini belirtti"

   In this example, the word "*düzenlenecek*" (literally "to be organized") is morphologically ambiguous. It can either be an adjective, or a verb.

2. Since we were dealing with newspaper articles, titles were also marked as sentences. It was very hard to determine the boundaries in such sentences. For example:

   "dış haberler servisi <**S**> iran ile taliban arasında iranlı diplomatların öldürülmesiyle başlayan gerginlik tırmanıyor"

In this example, the phrase *dış haberler servisi* (literaly "foreign news service") is a noun phrase, and there is no syntactic information (such as a final verb) in detecting this boundary.

3. According to our conventions, we did not mark sentence boundaries for the nested sentences inside a quoted piece of text. It was very hard without punctuation even for humans to decide sentence boundaries in such cases. This led to errors of the type:

   "purşasb talibana saldırıya niyetli misiniz sorusuna taliban bu kadar güçle saldırmak için çok küçük <**S**> kazandığı yerleri savaşarak değil hileyle elde etti <**S**> eğer bir gün biz onun kulağını çekmek istersek buna tamamen hazırız <**S**> taliban bize saldırırsa şiddetli bir cevap veririz şeklinde yanıt verdi"

5.5 Conclusion

We have presented a probabilistic model for automatically segmenting Turkish text into sentences when there is no case or punctuation information. We have tried different approaches to model sentence boundaries so that we can overcome the problems arising from the agglutinative nature of Turkish. First, we tried a word-based model, using only the surface forms of the words, then we have modeled the words according to the final inflectional groups of their morphological analyses. We have shown that the morphological model performs similarly with word-based model trained with 18 times more data. Furthermore, we obtained better performance when we combine these two models.

This system can be used as a preliminary step for processing the speech recognizer output for Turkish, or any other agglutinative languages. In that case, it may be possible to model prosodic information and augment it to our system, since prosody has been shown to be very effective for sentence segmentation of English.

Chapter 6

Topic Segmentation

6.1 Introduction

Topic segmentation is the task of automatically dividing a stream of text or speech into topically homogeneous blocks. Given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics change.

Topic segmentation is an important task for various language understanding applications, such as information extraction and retrieval (IR), and text summarization. An application may be as follows: Given a corpus of newspaper articles strung together, and a user's query, return a collection of coherent segments matching the query. Lacking a tool for detecting topic breaks, an IR application may be able to locate positions in its database, but be unable to determine how much of the surrounding data to provide to the user. Another example may be the broadcast news, or video-on-demand applications. There is no mark-up to indicate the topic boundaries and even the sentence boundaries in broadcast news. Also, segmenting text along topic boundaries may be useful for text summarization and anaphora resolution [Kozima, 1993].

Figure 6.1 gives an example of a topic change boundary from a broadcast

...tens of thousands of people are homeless in northern china tonight after a powerful earthquake hit an earthquake registering six point two on the richter scale at least forty seven people are dead few pictures available from the region but we do know temperatures there will be very cold tonight minus seven degrees <**TOPIC_CHANGE**> peace talks expected to resume on monday in belfast northern ireland former u. s. senator george mitchell is representing u. s. interests in the talks but it is another american center senator rather who was the focus of attention in northern ireland today here's a. b. c.'s richard gizbert the senator from america's best known irish catholic family is in northern ireland today to talk about peace and reconciliation a peace process does not mean asking unionists or nationalists to change or discard their identity or aspirations ...

Figure 6.1: An example of a topic boundary in a broadcast news word transcript.

news transcript. A corresponding Turkish example is given in Figure 6.2.

There has recently been increased interest in segmenting such information streams into topics. In 1997, the U.S. Defense Advanced Research Projects Agency (DARPA) initiated the Topic Detection and Tracking (TDT) Program [Allan *et al.*, 1998]. The purpose of this effort is to advance and accurately measure the state of the art in TDT and to assess the technical challenges to be overcome. This program consists of three major tasks:

1. **Topic Segmentation**: segmenting a stream of data, especially recognized speech, into distinct stories;
2. **Topic Detection**: identifying those news stories that are the first to discuss a new event occuring in the news; and
3. **Topic Tracking**: given a small number of sample news stories about an event, finding all following stories in the stream.

Topic segmentation is therefore also an *enabling* technology for other applications, such as tracking and new event detection.

In the next section, we review previous work on topic segmentation. In Section 6.3, we describe our morphological and lexical models as well as methods for

...Toshack eğer taşları yerinden oynatmazsa , yani çılgınlık yapmazsa Beşiktaş'ı şampiyonluğun adayları arasında göstermiştik
Ama adamın yapısı belli
Toshack ikinci yarıda Ertuğrul'u oyundan alıp , yerine Nihat'ı sokarken , oyun düzeninde değişikliğe gitti ve 3 - 5 - 2'ye döndü
Oktay'ı Ohen'in yanına çekti
Nitekim bu değişiklik biraz olsun Kartal'ı hem baskıdan kurtardı , hem de rakip savunmada çoğalmayı sağladı
Ne olduysa Oktay'ın inanılmaz golünden sonra oldu
Ardından Bursa panik yaptı , Nihat ustalık dolu bir vuruşla beraberliği sağladı
Beşiktaş ucuz kurtuldu
Ama her zaman böyle hata yaparsa yine kurtulabilir mi
<TOPIC_CHANGE>
ENERJİ Zirvesi'nin onur konuğu ABD eski Başkanı George Bush , dünyanın refahı için global projelerde birleşmenin şart olduğuna dikkat çekti
Dünya Enerji Konseyi'nin 17. Kongresi , 100'e yakın ülkeden 6 bine yakın uzman , teknisyen , işadamı ve siyasetçinin katılımı ile görkemli bir törenle başladı
ABD'nin Teksas eyaletinin Houston kentinde 13 -18 Eylül tarihleri arasında gerçekleştirilecek " Enerji ve Teknoloji : Gelecek 1000 Yıla Girerken Dünya Kalkınmasını Sürdürme " konulu kongrenin açılışını ABD Enerji Bakanı Bill Rihardson yaptı
Kongre'nin onur konuğu olan eski ABD başkanı George Bush açılışta yaptığı konuşmada , hükümetlerin temel görevlerinin dünya halklarının refahını arttırmak olduğunu belirterek , bu refah artışında dengeli çevre faktörü ve kaynaklarının akılcı kullanımını hesaba katan bir enerji politasının büyük önem taşıdığını söyledi
Dünya Enerji Konseyi'nin 17. Kongresi'nin açılış törenine Türkiye'den Enerji ve Tabii Kaynaklar Bakanı Cumhur Ersümer katıldı
Ersümer , Unocal Petrol Şirketi'nin sponsorluğunda düzenlenen at yarışını kazanan yarışçıya da kupasını verdi
<TOPIC_CHANGE>
Milli Eğitim Bakanı Hikmet Uluğbay , Başbakan Yılmaz'ın üniversitelerdeki türban esnekliği getirilmesine ilişkin sözlerini , 'beyanı Sayın Başbakanın ağzından duymadığım sürece fikir söyleyemem " diye değerlendirdi
Uluğbay , demokrasilerin bir kurallar rejimi olduğunu belirterek , kurallara uyulmaması halinde anarşi doğacağını söyledi
MİLLİ Eğitim Bakanı Hikmet Uluğbay , türban konusunda göreve geldiği günden bu yana sürdürdüğü tavrından " taviz'de bulunmayacağı mesajını verdi
Türbanlı öğretmenlerin görev yerinin değiştirilmesi ile ilgili olarak Uluğbay , herkesin kurallara uymak durumunda olduğunu aksi taktirde anarşi doğacağını ifade etti ...

Figure 6.2: An example of a topic boundary in a Turkish newspaper.

combining them. Section 6.4 reports our experimental procedures and results. We close with some general conclusions.

6.2 Previous Work

Prior work on topic segmentation is based on two broad classes of cues. On the one hand, one can exploit the fact that topics are correlated with *topical content-word usage*, and that global shifts in word usage are indicative of changes in topic. Quite independently, *discourse cues*, or linguistic devices such as discourse markers, cue phrases, syntactic constructions, and prosodic signals are employed by speakers (or writers) as generic indicators of endings or beginnings of topical segments. Interestingly, most previous work has explored either one or the other type of cue, but only rarely both. In automatic segmentation systems, word usage cues are often captured by statistical language modeling and information retrieval techniques. Discourse cues, on the other hand, are typically modeled with rule-based approaches or classifiers derived by machine-learning techniques (such as decision trees).

6.2.1 Approaches based on word usage

Most automatic topic segmentation work based on text sources has explored topical word usage cues in one form or other.

Kozima [1993] used mutual similarity of words in a sequence of text as an indicator of text structure. A text segment is a coherent scene if the words in that segment are linked via lexical cohesion relations. The lexical cohesion profile (LCP) records this mutual similarity of words in a sequence of text. Hills and valleys of the LCP closely correlate with changing of segments. In order to demonstrate the word similarity (σ) notion, consider the following examples:

$\sigma(cat, pet) = 0.133722$

$\sigma(cat, hat) = 0.001784$

Reynar [1994; 1998; 1999] presented a method which finds topically similar regions in the text by graphically modeling the distribution of word repetitions. This method is loosely based on dotplotting, a graphical technique described by Church [1993]. The application of this technique to text structuring uses word repetition information to divide a text into those regions determined to be most coherent by an optimization algorithm. The method has been successfully used to "discover" the document boundaries in concatenations of Wall Street Journal articles.

Hearst [1994; 1997] uses cosine similarity in a word vector space as an indicator of topic similarity. This algorithm, called TextTiling, is a simple, domain independent technique, that assigns a score to each topic boundary candidate (sentence boundaries). Topic boundaries are placed at the locations of valleys in this measure.

Several of the participating systems of the TDT-Pilot Program rely essentially on word usage: Yamron *et al.* [1998] model topics with unigram language models and their sequential structure with hidden Markov models (HMMs). The overall structure of the model is that of an HMM [Rabiner and Juang, 1986] in which the states correspond to topic clusters T_j, and the observations are sentences (or chopped units) $W_1, \ldots, W_N$. The resulting HMM, depicted in Figure 6.3, forms a complete graph, allowing for transitions between any two topic clusters. The exact number of topic clusters is not crucial, as long as it is large enough to make two adjacent topics in the same cluster unlikely. The observation likelihoods for the HMM states, $P(W_i|T_j)$, represent the probability of generating a given sentence W_i in a particular topic cluster T_j. 100 topic cluster LMs are automatically constructed, using the multipass k-means algorithm [Hartigan and Wong, 1979]. In this algorithm, at any given point there are k clusters. Initially stories are assigned to k clusters randomly, then for each story, the algorithm determines its distance to the closest cluster (based on the measure described below), and if this distance is below a threshold, inserts the story into that cluster and updates the statistics, otherwise creates a new cluster. This iteration is repeated until each

story is fine with its cluster.

The distance measure used in the clustering is a variation of the symmetric Kullback-Leibler (KL) metric [Kullback and Leibler, 1951]:

$$d = \sum_n (s_n/S) log \frac{s_n/S}{(c_n + s_n)/(C + S)} + \sum_n (c_n/C) log \frac{c_n/C}{(c_n + s_n)/(C + S)}$$

where s_n and c_n are the story and cluster counts for word w_n with $S = \sum s_n$ and $C = \sum c_n$.

Since the HMM emissions are meant to model the topical usage of words, but not topic-specific syntactic structures, the LMs consist of unigram distributions that exclude stop words (high-frequency function and closed-class words). To account for unobserved words they interpolate the topic cluster-specific LMs with the global unigram LM obtained from the entire training data. The observation likelihoods of the HMM states are then computed from these smoothed unigram LMs. We have tried to smooth the individual topic unigrams, but saw that our performance decreased as we expected, because, we want out language models to be specific to only one set of words, not all of them.

All HMM transitions within the same topic cluster are given probability one, whereas all transitions between topics are set to a global *topic switch penalty* (TSP) which is optimized on held-out training data. The TSP parameter allows trading off between false alarms and misses. Once the HMM is trained, they use the Viterbi algorithm [Viterbi, 1967] to search for the best state sequence and corresponding segmentation. Note that the transition probabilities in the model are not normalized to sum to one; this is convenient and permissible since the output of the Viterbi algorithm depends only on the relative weight of the transition weights.

Ponte and Croft [1997] extract related word sets for topic segments with the information retrieval technique of local context analysis (LCA), and then compare the expanded word sets. Each sentence of the text is run as a query against the LCA database and the top 100 concepts are returned. The original sentence

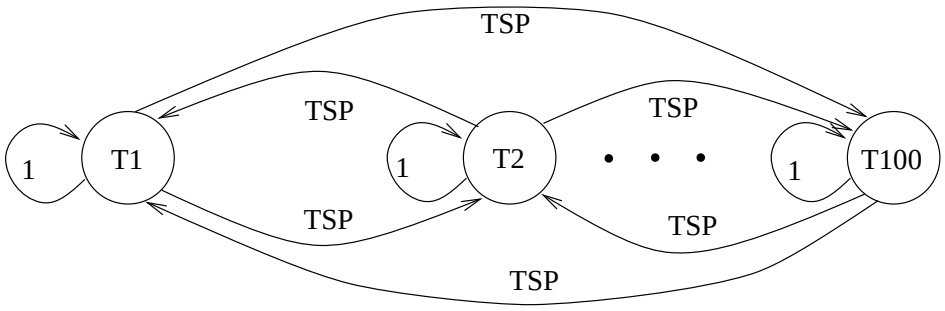

Figure 6.3: Structure of the basic HMM developed by Dragon for the TDT Pilot Project. The labels on the arrows indicate the transition probabilities. TSP represents the topic switch penalty.

is then replaced with the LCA concepts and the effect is that sentences which originally have few or perhaps no words in common will typically have many LCA concepts in common. Then it looks at the changes in the vocabulary. It is in fact similar to the topic models used in Dragon's model [Yamron *et al.*, 1998] and the discourse cue-words in CMU's method [Beeferman *et al.*, 1999].

6.2.2 Approaches based on discourse and combined cues

Previous work on both text and speech has found that cue phrases or discourse particles (items such as "now" or "by the way"), as well as other lexical cues, can provide valuable indicators of structural units in discourse [Grosz and Sidner, 1986; Passonneau and Litman, 1997, among others].

The UMass "HMM" approach described in the TDT Pilot Study Report [1998] uses an HMM that models the initial, middle, and final sentences of a topic segment, capitalizing on discourse cue words that indicate beginnings and ends of segments. Aligning the HMM to the data amounts to segmenting it. This approach may rely on the similarity of the training data to the test data somewhat heavily. Still, it shows that very simple discourse modeling can provide useful information.

At CMU, Beeferman *et al.* [1999] combined a large set of automatically selected lexical discourse cues in a maximum-entropy model. They also incorporated topical word-usage into the model by building two statistical language models: one static (topic-independent) and one that adapts its word predictions based on past words. They showed that the log likelihood ratio of the two predictors behaves as an indicator of topic boundaries, and can thus be used as an additional feature in the exponential model classifier. They had the best performance in the TDT-2 Study with 14.42% error rate.

IBM's approach for topic segmentation is a two stage process [Dharanipragada *et al.*, 1999]: In the first stage, the system uses a binary decision tree based on a probabilistic model to compute the probability of a boundary at every point in the automatic speech recognizer (ASR) transcript that has been labeled a non-speech event (such as pauses). In the second stage, they remove some of them in order to reduce the false alarm rate. This stage uses document-document similarity score to determine if adjacent stories are similar topically, and reject the hypothesized boundary between them. Their error rate in the TDT-2 Study was 16.51%.

In our previous work, we have successfully combined lexical and prosodic cues for automatic topic segmentation of speech [Stolcke *et al.*, 1999; Tür *et al.*, 2000; Shriberg *et al.*, 2000; Hakkani-Tür *et al.*, 1999]. For modeling topic boundaries prosodically, we used a wide range of features that were automatically extracted from the data. We trained probabilistic decision trees to predict the boundary type. For modeling the lexical information, similar to the Dragon HMM segmentation approach [Yamron *et al.*, 1998; van Mulbregt *et al.*, 1998], we built an HMM in which the states correspond to the topic clusters and the observations are sentences (or chopped units). In order to incorporate the probabilities obtained from the prosodic model, we inserted a fictitious *boundary* observation between adjacent sentences, and modified their original HMM, and introduced two more "boundary" states. Between two sentences, the model must pass through one of the boundary states, denoting either the presence or absence of a topic boundary. Likelihoods for the boundary states are obtained from the prosodic model. We have also modeled topic-initial and final sentences and inserted two more states for such sentences. The final HMM is depicted in Figure 6.4. The resulting

model thus effectively combines the Dragon and UMass HMM topic segmentation approaches described in the TDT Pilot Study Report [1998]. In preliminary experiments, we observed a 5% relative reduction in segmentation error with initial and final states over the baseline HMM topology of Figure 6.3.

The model was evaluated on broadcast news speech, and found to give a competitive performance (around 14% error according to the weighted TDT2 segmentation cost metric). Notably, the segmentation accuracy of the prosodic model alone is competitive with a word-based segmenter, and a combined prosodic/lexical HMM achieves a substantial error reduction (24-27%) over the individual knowledge sources.

6.3 The Approach

Topic segmentation in the paradigm used in this study and others proceeds in two phases. In the first phase, the input is divided into contiguous strings of words assumed to belong to the same topic. We refer to this step as "chopping". For example, in textual input, like newspapers, the natural units for chopping are sentences (as can be inferred from punctuation and capitalization), since we can assume that topics do not change in mid-sentence. Similarly, it is sometimes assumed for topic-segmentation purposes that topics only change at paragraph boundaries [Hearst, 1997]. For continuous speech input, the choice of chopping criteria is less obvious, it can be arbitrarily complex from chopping using pause durations as in SRI's approach [Stolcke *et al.*, 1999; Tür *et al.*, 2000] to using a decision tree like IBM [Dharanipragada *et al.*, 1999]. Since we deal with textual input with punctuation, for the first phase, we use a simple rule-based sentence segmentor developed by Hakkani-Tür [2000].

In the second phase, the sentences are grouped into contiguous stretches belonging to one topic, i.e., the sentence boundaries are classified into "topic boundaries" and "nontopic boundaries".[1] Topic segmentation is thus reduced to a

[1] We do not consider the problem of detecting recurring, discontinuous instances of the same topic, a task known as "topic tracking" in the TDT paradigm [Doddington, 1998].

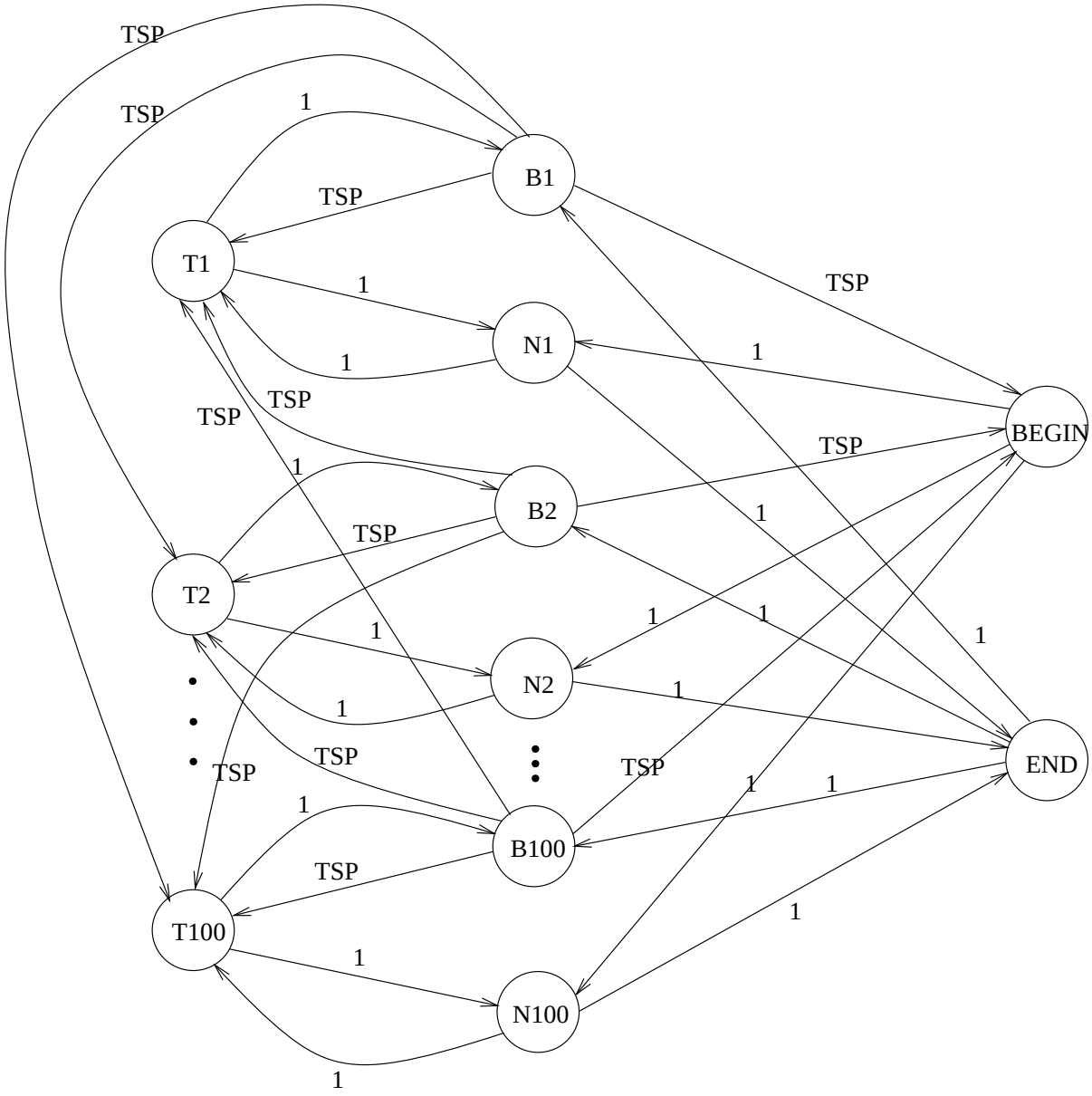

Figure 6.4: Structure of the final HMM with fictitious boundary states used for combining language and prosodic models. In the figure, states B1, B2, ..., B100 represent the presence of a topic boundary, whereas states N1, N2, ..., N100 represent topic-internal sentence boundaries. TSP is the topic switch penalty.

boundary classification problem. We will use B to denote the string of binary boundary classifications, and W to denote the word sequence. Our approach aims to find the segmentation B with highest probability given the information in W.

$$\underset{B}{\operatorname{argmax}}\, P(B|W) \tag{6.1}$$

using statistical modeling techniques.

For the second phase, we used an extension of Dragon's system, explained in the Section 6.2. In Dragon's system, lexical information is captured by statistical language models (LMs) embedded in a hidden Markov model [Yamron *et al.*, 1998; van Mulbregt *et al.*, 1998]. We preserved the HMM structure, in which states correspond to topic clusters T_j and the observations are sentences $W_1, \ldots, W_N$, as given in Figure 6.3. In their scheme, the observation likelihoods for the HMM states, $P(W_i|T_j)$, are obtained from the corresponding topic cluster language models, as described in Section 6.2. This approach was based purely on topical word distributions. We extend it to also handle morphological aspects of Turkish, using stems of the words and then using only nouns in forming the topic clusters, as described in the following subsections.

6.3.1 Word-based Modeling

In order to gauge our baseline performance, similar to Dragon, we automatically constructed 100 topic cluster LMs, using the multipass k-means algorithm described in [Yamron *et al.*, 1998]. Since the HMM emissions are meant to model the topical usage of words, but not topic-specific syntactic structures, the LMs consist of unigram distributions that exclude stop words (high-frequency function and closed-class words)[2]. To account for unobserved words we interpolate the topic cluster-specific LMs with the global unigram LM obtained from the entire training data. The observation likelihoods of the HMM states are then computed from these smoothed unigram LMs.

[2]See Appendix B for a list of stopwords.

Word	Freq	Meaning
gol	1222	goal
ikinci	912	second
Beşiktaş	867	Beşiktaş
teknik	781	technical
Galatasaray	773	Galatasaray
Fenerbahçe	699	Fenerbahçe
orta	678	middle
takım	665	team
dk.	655	min.
sarı	622	yellow
maç	592	match
yarıda	575	half *Loc*
top	521	ball
Trabzonspor	479	Trabzonspor
yaptı	473	did
Mehmet	471	Mehmet
Hakan	462	Hakan
dakikada	450	minute *Loc*
maçı	449	match *Acc*
futbol	445	football
Fatih	413	Fatih
yarı	412	half
oyun	406	game
Ali	384	Ali

Table 6.1: The most frequent words in one of the clusters, containing mostly football news articles. *Loc* denotes locative case, *Acc* denotes accusative case.

Table 6.1 gives a list of the most frequent words in the same topic cluster, containing mostly football news articles. *Beşiktaş*, *Galatasaray*, *Fenerbahçe*, and *Trabzonspor* are top Turkish football teams, *Hakan*, *Mehmet*, and *Ali* are the top players, and *Fatih Terim* is the trainer of *Galatasaray*.

6.3.2 Stem-based Modeling

Word-based modeling works well in languages in which there is very little or no morphology, such as English. On the other hand, morphologically rich languages,

like Turkish, suffer from the fact that the number of word forms one can derive from a Turkish root form may be in the millions [Hankamer, 1989]. Because of this reason, the number of distinct word forms is much larger than that of English.

More specifically, word-based approach suffers from this characteristic of Turkish in two major ways:

1. Using the surface forms of the words results in data sparseness in the training data. When we consider the words with different inflectional and derivational suffixes different, then we have to deal with data sparseness.

   Table 6.2 gives a list of 26 different word forms involving the stem *gol* (goal), in the cluster mentioned in Table 6.1. The meaning of the features of the morphological analyses are provided in Appendix A.

   This sparseness does not only badly damage the quality of the language models, but also the performance of the clustering algorithm. Since we check for the similarity distance of a given document and a cluster, and use the words themselves in this computation, the result may be misleading while using the words. So we can expect a better clustering using stemming beforehand.

2. The second drawback of using a word-based model is that, while segmenting, using the surface forms of the words leads to a lower performance because of two reasons.

   - The first reason is that a word with an unseen inflectional or derivational form would not contribute to the statistical computation, although its stem may be in the vocabulary.
   - Even though a word is the language model of a cluster, the probability assigned to it may be misleading.

It is clear that, removing the suffixes the words, and using the root words will prevent the data sparseness, and the unigram language models obtained from

Word	Freq	Morphological Analysis
gol	1222	goal+Noun+A3sg+Pnon+Nom
golü	350	goal+Noun+A3sg+Pnon+Acc or
		goal+Noun+A3sg+P3sg+Nom
gole	150	goal+Noun+A3sg+Pnon+Dat
golle	138	goal+Noun+A3sg+Pnon+Ins
goller	126	goal+Noun+A3pl+Pnon+Nom
golde	85	goal+Noun+A3sg+Pnon+Loc
golün	75	goal+Noun+A3sg+Pnon+Gen or
		goal+Noun+A3sg+P2sg+Nom
golünü	63	goal+Noun+A3sg+P3sg+Acc or
		goal+Noun+A3sg+P2sg+Acc
golüyle	62	goal+Noun+A3sg+P3sg+Ins
golcü	59	goal+Noun+A3sg+Pnon+Nom^DB+Adj+Agt
golleri	48	goal+Noun+A3pl+P3sg+Nom or
		goal+Noun+A3pl+Pnon+Acc or
		goal+Noun+A3pl+P3pl+Nom or
		goal+Noun+A3sg+P3pl+Nom
golden	45	goal+Noun+A3sg+Pnon+Abl
gollerle	40	goal+Noun+A3pl+Pnon+Ins
gollük	37	goal+Noun+A3sg+Pnon+Nom^DB+Adj+FitFor
gollü	26	goal+Noun+A3sg+Pnon+Nom^DB+Adj+With
golüne	24	goal+Noun+A3sg+P3sg+Dat or
		goal+Noun+A3sg+P2sg+Dat
golleriyle	20	goal+Noun+A3pl+P3sg+Ins or
		goal+Noun+A3pl+P3pl+Ins or
		goal+Noun+A3sg+P3pl+Ins
golsüz	18	goal+Noun+A3sg+Pnon+Nom^DB+Adj+Without
golcüsü	18	goal+Noun+A3sg+Pnon+Nom^DB+Noun+Agt+A3sg+P3sg+Nom
golünde	16	goal+Noun+A3sg+P3sg+Loc or
		goal+Noun+A3sg+P2sg+Loc
gollerde	15	goal+Noun+A3pl+Pnon+Loc
goldeki	15	goal+Noun+A3sg+Pnon+Loc^DB+Det
gollerin	12	goal+Noun+A3pl+Pnon+Gen or
		goal+Noun+A3pl+P2sg+Nom
golünden	10	goal+Noun+A3sg+P3sg+Abl or
		goal+Noun+A3sg+P2sg+Abl
gollerini	9	goal+Noun+A3pl+P3sg+Acc or
		goal+Noun+A3pl+P2sg+Acc or
		goal+Noun+A3pl+P3pl+Acc or
		goal+Noun+A3sg+P3pl+Acc
gollere	8	goal+Noun+A3pl+Pnon+Dat

Table 6.2: The frequency table for the root word *gol* (goal) in the cluster mentioned in Table 6.1.

the topic clusters would be more effective. So we decided to use the root words instead of the surface forms of the words, and build stem-based language models, instead of word-based language models.

In order to do this, we used a preprocessing module, developed by Hakkani-Tür [2000], which tokenizes the training data, analyzes the tokens using the morphological analyzer developed by Oflazer [1993], groups the collocations, and finally removes some obviously improbable morphological parses in order to reduce the morphological ambiguity. Then, we extracted the roots of the words, and rebuilt the training corpus using only these roots. When there were more than one root for a word, because of the morphological ambiguity, we used all of the roots. However, this root ambiguity was not a real problem as there were only 1.15 distinct roots per word on the average.

As expected, we obtained clusters with smaller number of root words, and each with higher frequencies. Table 6.3 lists the most frequent root words in corresponding cluster containing mostly football news.

6.3.3 Noun-based Modeling

When we analyzed Table 6.3, and other clusters, we saw that in order to model the topical usage of words, it was not enough to exclude the stopwords. In fact, only nouns would be sufficient to model the topics. Since we have the morphological analyses of the words, it was straightforward for us to test this hypothesis.

Instead of using the stems of words, we only used the stems of the morphological parses that have a noun root form. After using the same clustering algorithm, we ended up with new clusters. The most frequent nouns for the cluster containing mostly football related articles is listed in Table 6.4. Common verbs such as, *ol* (be) , *al* (take), *yap* (make), and *et* (do) and somewhat football related verbs, such as *oyna* (play), *çık* (exit), and *at* (score) disappeared in Table 6.4 when we compare with Table 6.3.

[3]Note that the frequent word *oyun* (game) has another morphological parse, meaning "your vote", hence the appearance of the root *oy* (vote).

Word	Freq	Meaning
gol	2271	goal
maç	2048	match
oyun	1781	game
takım	1382	team
ol	1317	be
oy[3]	1273	vote
al	1264	take
top	1228	ball
futbol	1227	football
oyna	1224	play
yap	1219	do or make
yarı	1101	half
Galatasaray	1018	Galatasaray
saha	996	field
Hakan	986	Hakan
Beşiktaş	974	Beşiktaş
at	948	throw
dakika	892	minute
Fenerbahçe	872	Fenerbahçe
rakip	866	opponent
çık	826	exit
orta	785	middle
et	755	do or make
ikinci	734	second

Table 6.3: The most frequent stems in a cluster, containing mostly football news articles.

Word	Freq	Meaning
gol	2562	goal
maç	2412	match
oyun	2071	game
takım	1659	team
futbol	1492	football
oy	1429	vote
yarı	1275	half
top	1257	ball
Galatasaray	1230	Galatasaray
Beşiktaş	1201	Beşiktaş
saha	1189	field
Fenerbahçe	1162	Fenerbahçe
dakika	1029	minute
orta	868	middle
rakip	852	opponent
lig	695	league
kale	657	goal
dk.	642	min.
pozisyon	638	position
hata	606	error
teknik	594	technical
Hakan	579	Hakan
hakem	543	referee
alan	541	space or field

Table 6.4: The most frequent nouns in a cluster, containing mostly football news articles.

6.4 Experiments and Results

To evaluate our topic segmentation models we carried out experiments in the TDT paradigm. We first describe our training and test data, then give results obtained with the baseline word-based, stem-based, and noun-based language models. We used SRILM toolkit for language modeling and decoding [Stolcke, 1999]. In our work, we assumed that each news piece contains only one topic, and attempted to find out article boundaries. Hand-checking of a subset of articles showed that this assumption was true except for a few cases.

6.4.1 Training Data

Topic unigram language models were trained on texts extracted from the web resources of Milliyet newspaper, covering the period from January 1, 1997 through September 12, 1998. For training the language models, we removed stories with fewer than 300 and more than 3,000 words, leaving 14,495 stories with an average length of 432 words, 500 stems, or 310 nouns, excluding stop words[4], for a total of 376,371 distinct words, 128,125 distinct stems, or 119,475 distinct nouns.

6.4.2 Test Data

We evaluated our system on a test set of 100 news articles, covering the period from September 12, 1998 through September 14, 1998, comprising 2,803 sentences, 32,772 words, 38,329 stems, or 24,807 nouns, excluding stopwords. The topic switch penalty was optimized on the development set of 99 news articles from the same newspaper, between September 14, 1998 and September 16, 1998, comprising 3,180 sentences, 33,728 words, 39,106 stems, or 25,615 nouns, excluding stopwords.

[4]See Appendix B for a list of stopwords.

6.4.3 Evaluation metrics

We have adopted the evaluation paradigm used by the TDT2—Topic Detection and Tracking Phase 2 [Doddington, 1998] program, allowing fair comparisons of various approaches both within this study and with respect to other recent work. Segmentation accuracy was measured using TDT evaluation software from NIST, which implements a variant of an evaluation metric suggested by Beeferman *et al.* [1999].

The TDT segmentation metric is different from those used in most previous topic segmentation work, and therefore needs some discussion. It is designed to work on data streams in the absence of a small set of potential topic boundaries given *a priori*, such as paragraph or sentence boundaries. It also gives proper partial credit to segmentation decisions that are close to actual boundaries; for example, when a segmenter places a boundary one word from an actual boundary that is considered a lesser error than when the hypothesized boundary is off by, say, 100 words.

The evaluation metric reflects the probability that two positions in the corpus probed at random and separated by a distance of k words are correctly classified as belonging to the same story or not. If the two words belong to the same topic segment, but are erroneously claimed to be in different topic segments by the segmenter, then this will increase the system's *false alarm* probability. Conversely, if the two words are in different topic segments, but are erroneously marked to be in the same segment, this will contribute to the *miss* probability. The false alarm and miss rates are defined as averages over all possible probe positions with distance k. In the TDT-2 project, k is a constant and equals to 50.

Formally, miss and false alarm rates are computed as

$$P_{Miss} = \frac{\sum_s \sum_{i=1}^{N_s-k} d^s_{hyp}(i, i+k) \times (1 - d^s_{ref}(i, i+k))}{\sum_s \sum_{i=1}^{N_s-k} (1 - d^s_{ref}(i, i+k))} \quad (6.2)$$

$$P_{FalseAlarm} = \frac{\sum_s \sum_{i=1}^{N_s-k} (1 - d^s_{hyp}(i, i+k)) \times d^s_{ref}(i, i+k)}{\sum_s \sum_{i=1}^{N_s-k} d^s_{ref}(i, i+k)} \quad (6.3)$$

where the summation is over all broadcast news s and word positions i in the test corpus and where

$$d^s_{sys}(i,j) = \begin{cases} 1 & \text{if words } i \text{ and } j \text{ in news } s \text{ are deemed by} \\ & sys \text{ to be within the same story} \\ 0 & \text{otherwise} \end{cases}$$

Here *sys* can be *ref* to denote the reference (correct) segmentation, or *hyp* to denote the segmenter's decision.

We used the same parameters as used in the official TDT2 evaluation. Furthermore, again following NIST's evaluation procedure, we combine miss and false alarm rates into a single *segmentation cost* metric

$$C_{Seg} = C_{Miss} \times P_{Miss} \times P_{seg} + C_{FalseAlarm} \times P_{FalseAlarm} \times (1 - P_{seg}) \qquad (6.4)$$

where the $C_{Miss} = 1$ is the cost of a miss, $C_{FalseAlarm} = 1$ is the cost of a false alarm, and $P_{Seg} = 0.3$ is the *a priori* probability of a segment being within an interval of k words on the TDT2 training corpus.

Another parameter in the NIST evaluation is the deferral period, i.e., the amount of look-ahead before a segmentation decision is made. In all our experiments we allowed unlimited deferral, effectively until the end of the news show being processed.

6.4.4 Segmentation Results

Table 6.5 shows the results of the Turkish topic segmenter, using word-based, stem-based, and noun-based approaches.

These results are consistent with our intuition, that we have tried to explain in the previous section. As expected, the word-based model suffered from data sparseness, and 28.61% improvement is achieved for the development set when we use the stems of the words. Furthermore, it is possible to obtain an additional

Model	Development Set			Test Set		
	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}
Chance	1.0000	0.0000	0.3000	1.0000	0.0000	0.3000
Human Performance	0.2093	0.0176	0.0742	N/A	N/A	N/A
Word-based	0.4394	0.0658	0.1779	0.3560	0.0752	0.1594
Word-based (Random)	0.3412	0.0286	0.1224	0.3840	0.0427	0.1451
Stem-based	0.2704	0.0655	0.1270	0.2552	0.0708	0.1261
Noun-based	**0.2627**	**0.0413**	**0.1077**	**0.2487**	**0.0492**	**0.1090**

Table 6.5: Summary of error rates with different language models. A "chance" classifier that labels all potential boundaries as non-topic would achieve 0.3 weighted segmentation cost. "Random" indicates that the articles are shuffled.

15.19% improvement using only nouns, achieving a total of 39.46% improvement over our baseline word-based model. For the test set, the results are also similar, and we achieved 20.89% improvement when we used the stem-based approach, and our results are 31.61% better when we used the noun-based approach.

Comparing these three modeling approaches, we observe that stem-based and noun-based models have a 38%-40% lower miss probability than the word-based model in the development data. This rate is 28%-30% in the test set. This enormous decrease in the miss probability is the main reason of the final improvement. We would say that, using stems, we have obtained more discriminative topic unigram language models in the clustering phase, hence we have missed fewer topic boundaries. Additionally, when we have used the noun-based models, we see that there is a 31%-37% improvement over the stem-based models in the false alarm probabilities.

Let us analyze these results using a concrete example. Consider the following sentence from an article on football: *Son dakikalarda Galatasaray'ın atakları sıklaştı, Hakan attığı golle ağları sarstı.* (Literally, "In the last minutes, Galatasaray's attacks became more frequent, Hakan shook the net with the goal, he scored.") Table 6.6 shows the individual unigram probabilities of the words in a cluster including mainly football news articles for both word-based and stem-based approaches. Note that, due to data sparseness, all of these words, though related with football have less probability when compared to stem-based and

Word	Morphological Analysis	Word-based Probability	Stem-based Probability	Noun-based Probability
Son	Last+Adj	0	0	0
dakikalarda	minute+Noun+A3pl+Pnon+Loc	0.000337	0.004930	0.007296
Galatasaray'ın	Galatasaray+Noun+Prop+A3sg+Pnon+Gen	0.001433	0.005598	0.008679
atakları	attack+Noun+A3pl+P3sg+Nom	0.000072	0.001192	0.001600
sıklaştı	frequent+Adj^DB+Verb+Become+Pos+Past+A3sg	0	0.000557	0
Hakan	Hakan+Noun+Prop+A3sg+Pnon+Nom	0.002556	0.005422	0.004087
attığı	score+Verb+Pos^DB+Adj+PastPart+P3sg	0.001232	0.005458	0
golle	goal+Noun+A3sg+Pnon+Ins	0.000760	0.012454	0.018019
ağları	net+Noun+A3pl+Pnon+Acc	0.000138	0.000428	0.000595
sarstı	shake+Verb+Pos+Past+A3sg	0.000001	0.000127	0

Table 6.6: The unigram probabilities of the words in the example sentence. Note that, the word *son* (last) is a stopword, hence gets 0 probability.

noun-based models. Furthermore, the word *sıklaştı* (became frequent) received 0 probability, since its surface form is unseen in the training data, although its stem *sık* (frequent) gets some probability.

6.4.5 Error Analysis

When we analyze our errors, we see that errors are made when there are topically very similar news articles in a sequence, or when an article contains more than one topic, though this second case is less likely. This is why we obtained better performance on the test set than the development set for both word-based and stem-based models, although we set the topic switch penalty on the development set. When we analyzed this, we see that development set is *harder* to segment than the test set, in the sense that it includes articles with very similar consecutive topics. Note that, because of this, the miss probability of a human annotator is about 20%. When we ordered the articles randomly, this difference disappeared.

6.4.6 Results Compared to Topic Segmentation of English

It would be useful to provide word-based segmentation error rates obtained from a recent work [Tür *et al.*, 2000] for English Broadcast News corpus. As shown

Corpus	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}
Turkish	0.4394	0.0658	0.1779
English	0.4685	0.0817	0.1978

Table 6.7: Word-based segmentation error rates for English and Turkish corpora.

in Table 6.7, the two test sets have comparable behavior. Stem-based and noun-based models are not available for English. It would be interesting to try these approaches for English, too.

6.4.7 False Alarm vs. Miss Rates

The trade-off between false alarms and miss probabilities is shown in more detail in Figure 6.5, which plots the two error metrics against each other. Note that the false alarm rate does not reach 1 because the segmenter is constrained by the chopping assumption: Topic boundaries only exist on the sentence boundaries. According to this graph, the decrease in the false alarm rate is steeper when we use the stems of words and nouns instead of the surface forms of the words, hence it is possible to obtain lower miss rates. For example, in order to obtain a 0.07 false alarm rate, you have to accept a miss rate of 0.4 using the words, whereas this rate is only 0.2 when using the stems of the nouns.

6.4.8 The Effect of Chopping

In all experiments we have presented, we have used the actual sentence boundaries as our topic boundary candidates. In fact this is not the case, when we are dealing with speech recognizer output. For such a case, we have thought that we can use our sentence segmentor, we have presented in Chapter 5.

In order to see the effect of this assumption, we have performed two sets of experiments. In the first set, instead of using actual sentence boundaries, we have used fixed length sentences. In order to get satisfactory results, we have optimized

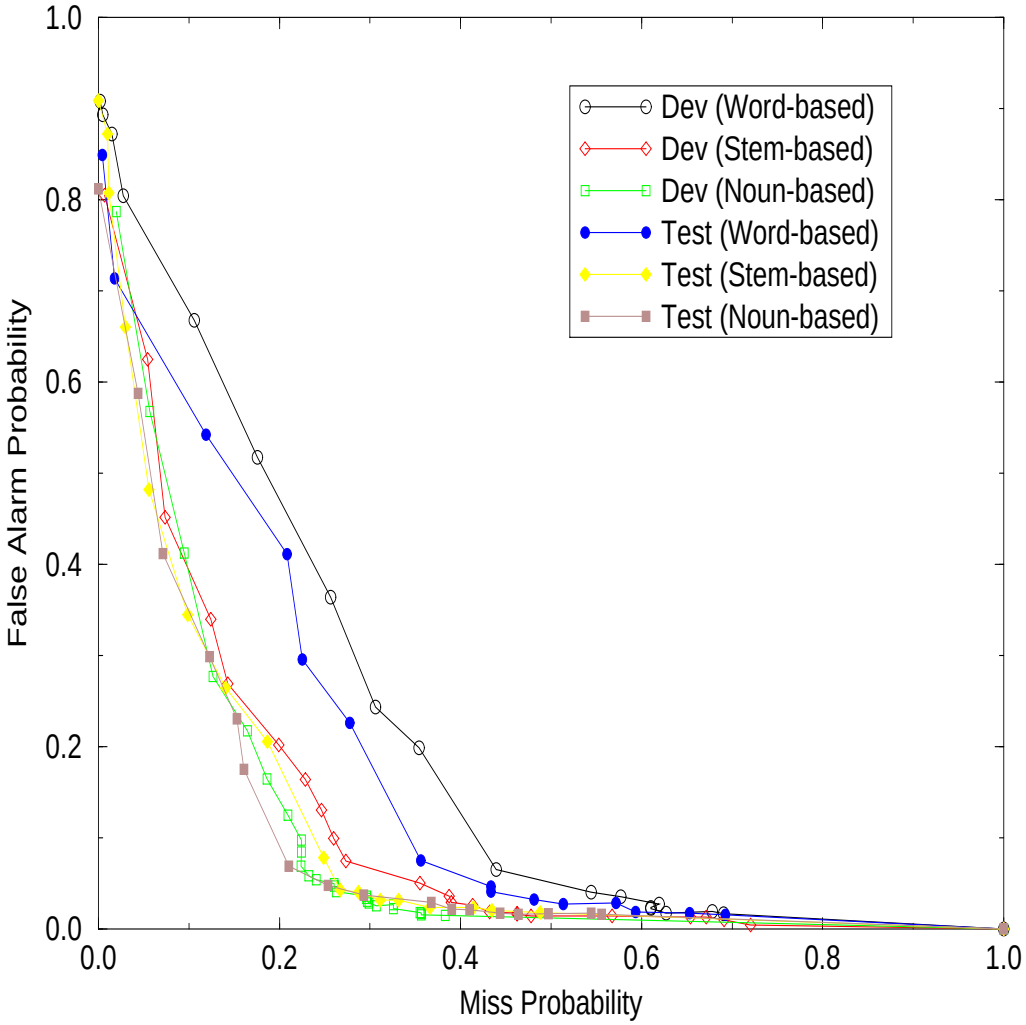

Figure 6.5: False alarm versus miss probabilities for automatic topic segmentation of news for both development (Dev) and test (Test) sets.

Chopping Criterion	Development Set			Test Set		
	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}	P_{Miss}	$P_{FalseAlarm}$	C_{Seg}
True Boundaries	0.4394	0.0658	0.1779	0.3560	0.0752	0.1594
Fixed Length (15 Words)	0.5884	0.0519	0.2128	0.5232	0.0763	0.2104
Sentence Segmentor	0.4890	0.0770	0.2006	0.5578	0.1032	0.2396

Table 6.8: Word-based segmentation error rates using word-based models, when we use fixed length sentences, or when we use the sentence boundaries marked by the automatic sentence segmentor, or when they are given (True Boundaries).

this length using our held-out data, and a length of 15 words has appeared to be optimum. As a second set of experiments, we have segmented our data, using the sentence segmentation system, we have defined in Chapter 5. For optimum segmentation performance, we have used both lexical and morphological models. Table 6.8 presents our results of these two sets of experiments. In both cases, for both development and test data, our performance decreased at least 10%. For the development set, automatic segmentation performed better than using fixed length sentences. When we analyze why we obtained worse results in the automatically segmented test data, we see that this set has longer sentences, most probably including more speech transcriptions, i.e. a combination of sub-sentences, hence more prone to segmentation errors. These errors propagate to topic segmentation errors.

6.5 Conclusion

We have presented a probabilistic model for automatically segmenting Turkish text into topically homogeneous blocks. We tried three different approaches to model topics so that we can overcome the problems arising from the agglutinative nature of Turkish. First, we tried a baseline model, using only the surface forms of the words, then we have modeled the stems of the words, and obtained a significant improvement. Finally we modeled only the stems of the nouns, and reached 10.90% segmentation error rate according to the weighted TDT2 segmentation cost metric on our test set, which was 32% better than the baseline model.

After this basic task, we are now ready to proceed to other more complicated TDT tasks, such as detection and tracking. Given this segmentation framework, it is straightforward for us to develop similar systems. For example, this same framework can be used for other clustering algorithms, such as instead of topic clustering, we can cluster writers. Also it may be interesting to see the performance of clustering algorithms other than the k-means clustering we used in this task.

One application of such a system is that, it is now possible to keep track of news articles, or any similar media. This algorithm does not use case or punctuation information, thus it can be extended to cover broadcast news, and even visual cues can be added in this segmentation task. Also it is possible to keep track of documents in the web. Consider a cite like Yahoo, which classifies the web sites according their contents. This classification can be done using a system similar to the one presented in this chapter. It may also be possible to think of hierarchical clusters in order to better use in such a task. Indeed, multilevel clustering may be effective in differentiating sub-topics in this task, too. For example, the topic clusters we use in this thesis are weak to differentiate the football news articles from the wrestingling news articles.

Chapter 7

Name Tagging

7.1 Introduction

One of the basic tasks in an information extraction system is marking names (persons, locations, and organizations), and certain structured expressions (monetary values, percentages, dates and times). This is known as *named entity (NE) extraction* task. In this task, finding only names is called *name tagging.*

Named entity extraction task has been introduced by DARPA, and evaluated as an understanding task in both the Sixth and Seventh Message Understanding Conferences (MUC-6 [1995] and MUC-7 [1998]). A very detailed definition of the named entity extraction task has been developed in the framework of these programs [Chinchor and Robinson, 1998].

We would like to first give the flavor of this task, and then define the task in detail, while mentioning some problems and difficulties of finding and tagging names in a text.

Name tagging task is limited to proper names, acronyms, and perhaps miscellaneous other unique identifiers, which are categorized via their type as follows [Chinchor and Robinson, 1998]:

```
...Good evening from <ENAMEX TYPE="LOCATION''>Havana </ENAMEX>
where one of the day's big stories has begun to unfold.
One of them the Pope is here and the world is waiting to
see whether he will shake up this island and the veteran
communist leader who runs it <ENAMEX TYPE=''PERSON''>Fidel
Castro</ENAMEX>.  The other very big story of the day is in
<ENAMEX TYPE="LOCATION''>Washington</ENAMEX> where the <ENAMEX
TYPE="ORGANIZATION''>White House</ENAMEX> administration
has already been badly shaken up by the possibility that
president <ENAMEX TYPE="PERSON''>Clinton</ENAMEX> and one
of his advisors <ENAMEX TYPE="PERSON''>Vernon Jordan</ENAMEX>
obstructed justice.  ...
```

Figure 7.1: An example of an example broadcast news word transcript, whose named entities are marked.

- **ORGANIZATION:** named corporate, governmental, or other organizational entity
- **PERSON:** named person or family
- **LOCATION:** name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.)

IE systems are usually evaluated and compared using broadcast news transcriptions, where there are lots of such entities. Consider Figure 7.1 for an example news piece where the names are marked in SGML tags.

Although name tagging seems like a very straigtforward process, even human annotators have a performance of 98%-99%. In real text, there are lots of cases in which it is very hard to determine the type of the name (for example determining whether *Washington* is a location or a person), or even whether it is a name or not (for example *Dow Jones* is not a name). This is why the official guideline of this task is very detailed, and tries to capture all kinds of such cases.

7.2 Task Definition

In the name tagging task, names are marked with the SGML tag "ENAMEX". The subcategorization is captured by a SGML tag attribute called TYPE, which is defined to be either "PERSON", "LOCATION", or "ORGANIZATION".

For all types of names, multi-word strings, containing name substrings are not decomposable.

"Arthur Anderson Consulting"
<ENAMEX TYPE="ORGANIZATION">Arthur Anderson Consulting</ENAMEX>
[no markup for "Arthur Anderson" alone]

"U.S. Fish and Wildlife Service"
<ENAMEX TYPE="ORGANIZATION">U.S. Fish and Wildlife Service</ENAMEX>
[no markup for "U.S." alone]

``North and South America''
<ENAMEX TYPE="LOCATION''>North and South America</ENAMEX>

In a genitive-possessive noun phrase construction, the possessor and possessed ENAMEX substrings should be tagged separately.

"Bilkent University's Graduate School of Business"
<ENAMEX TYPE="ORGANIZATION">Bilkent University</ENAMEX>'s <ENAMEX
TYPE="ORGANIZATION">Graduate School of Business</ENAMEX>

A very problematic case is tagging aliases, such as acronyms, nicknames, truncated names, certain proper metonyms[1].

"IBM" [alias for International Business Machines Corp.]
<ENAMEX TYPE="ORGANIZATION">IBM</ORGANIZATION>

"Big Blue" [alias for International Business Machines Corp.]

[1] *Metonym*: a figure of speech consisting of the use of the name of one thing for that of another of which it is an attribute or with which it is associated (as "crown" in "lands belonging to the crown")

```
<ENAMEX TYPE="ORGANIZATION">Big Blue</ORGANIZATION>

"Mr.  Fix-It" [nickname for candidate for head of the CIA]
Mr.  <ENAMEX TYPE="PERSON">Fix-It</ENAMEX>

"The Pentagon announced..."
The <ENAMEX TYPE="ORGANIZATION">Pentagon</ORGANIZATION> announced
...
```

But aliases that refer to broad industrial sectors, political power centers, etc., rather than to specific organizations are not to be tagged. For example, do not tag "Wall Street" as an alias for the U.S. stock market, "Japan Incorporated" as an alias for Japanese Industries, "Uncle Sam" and "Washington" as aliases for the U.S. government, or "Capitol Hill" as an alias for the Congress, since these do not refer to specific organizations. The "Ivy League" refers to a specific set of universities, but does not seem to be a specific organization in its own right. Similarly, the "Axis" (WWII Germany-Japan-Italy) and the "Iron Curtain countries" are aliases for finite sets of entities, but not for specific organizations with corporation-like infrastructures.

Metonyms, herein designated "common" metonyms, that reference political, military, athletic, and other organizations by the name of a city, country, or other associated location are not to be tagged as organization. In these cases, the association between the name's semantic type and the organization is sufficiently predictable and non-idiosyncratic as to preclude a dictionary gloss; hence the name should be tagged as a LOCATION. Some examples of "common" metonyms follow.

```
"Germany invaded Poland in 1939."
<ENAMEX TYPE="LOCATION">GERMANY</LOCATION> invaded ...

"Galatasaray defeated Kartal by a score of 2 to 1.
<ENAMEX TYPE="LOCATION">Galatasaray</LOCATION> defeated <ENAMEX TYPE="ORGANIZA
...
```

Note that links from LOCATION-tagged names to organizations (e.g.

"Galatasaray" to the "Galatarasay Sports Club" football team) are left to occur, along with anaphora-resolution, at a processing level higher than named entity tagging.

Miscellaneous types of proper names that are *not* to be tagged as ENAMEX include artifacts, other products, and plural names that do not identify a single, unique entity.

```
"Wall Street Journal"
[no markup]

"Dow Jones Industrial Average"
[no markup, not even for "Dow Jones"]

"Ford Taurus"
<ENAMEX TYPE="ORGANIZATION">Ford</ENAMEX> Taurus
```

7.2.1 Organizations

Miscellaneous types of proper names that are to be tagged as ORGANIZATION include stock exchanges, multinational organizations, political parties, orchestras, unions, embassies, factories, hospitals, hotels, museums, universities, non-generic governmental entity names such as "Congress" or "Chamber of Deputies", sports teams and armies (unless designated only by country names, which are tagged as LOCATION). Generic entity names such as "the police" and "the government" are not to be tagged. An helpful criteria in deciding whether a name is an organization is to check whether there is an office space in it.

7.2.2 Locations

Examples of place-related strings that are tagged as LOCATION include named heavenly bodies, continents, countries, provinces, counties, cities, regions, districts, towns, villages, neighborhoods, airports, highways, street names, street

addresses, oceans, seas, straits, bays, channels, sounds, rivers, islands, lakes, national parks, mountains, fictional or mythical locations, and monumental structures, such as the Eiffel Tower and Washington Monument, that were built primarily as monuments. Note that airports are to be tagged as location, even though there are lots of office spaces in an airport.

```
"Mississippi River"
<ENAMEX TYPE="LOCATION">Mississippi River</ENAMEX>
(not:  <ENAMEX TYPE="LOCATION">Mississippi</ENAMEX> River)
```

7.2.3 Persons

Named person and families are to be marked as PERSON. Similar to other types, the longest name is to be tagged.

```
``Hillary and Bill Clinton''
<ENAMEX TYPE=''PERSON''>Hillary and Bill Clinton</ENAMEX>
```

7.3 Previous Work

Similar to most other language processing systems, developers have approached the named entity extraction problem as one of building a hand-crafted rule-based system, an automatically trained system, or a combination of these two approaches. Table 7.1 summarizes the performances of the systems participated in MUC-7. Annotators indicate the performance of human in this task. Note that, we are still far from the human performance.

In this section we are going to discuss some other systems, which did not participate in the MUC-7 conference, too, such as FASTUS of SRI.

Model	*F-Measure*[2]
Annotator 1	99.03%
Annotator 2	98.34%
HCRC-LTG	94.05%
IsoQuest	91.32%
BBN	90.61%
NYU	88.66%
U. of Manitoba	86.07%
U. of Sheffield	85.25%
MITRE	83.75%
Oki	81.13%
UMIST	79.50%
Kent Ridge	76.65%
U. of Durham	74.53%

Table 7.1: MUC-7 Name Tagging Scores for English.

7.3.1 Rule-based Approaches

Rule-based systems performed better than statistical systems in name tagging task. In this section, we will discuss UMIST's FACILE, IsoQuest's NetOwl, SRI's FASTUS, University of Durham's LOLITA, University of Sheffield's LASIE-II systems, and the Oki system from Japan.

FACILE

UMIST participated in the MUC-7 NE task in the framework of the FACILE project, co-funded by the European Community's Language Engineering program [Black *et al.*, 1998]. Their system is completely rule-based, and does not employ any kind of learning techniques in any phase.

The FACILE system first preprocesses input to the system, then recognizes special formatting, tokenizes, tags, looks up single and multi-word tokens in a

database, and carries out proper name recognition and classification. Furthermore, the rules can be assigned an explicit weight which is used in choosing between competing analyses. They have used over 100 rules of the form $a => B/D$, where A, B, C, and D are attribute operator value expressions. Let us demonstrate this with the following example, which captures university names as an organization. NP stands for noun phrase, and semantic information is obtained from a knowledge-base.

```
[syn=NP,sem=ORG] (0.9) =>
 \ [norm=''university''],
   [token=''of''],
   [sem=REGION|COUNTRY|CITY] / ;
```

The FACILE system, though had achieved about 92%-93% F-measure in the training data, did not perform equally good in the test data, and reached an F-measure of 79.50

NetOwl

The first commercial product, emerged from the named entity extraction task is called NetOwl Extractor from IsoQuest founded by SRA International [Krupka and Hausman, 1998]. This system is also rule-based, where rules consist of a pattern and an action. The pattern is similar to a regular expression and consists of special operator and operands that match portions of text. The action performs operations on the text, such as tagging name with a classification. The application of the rules needs a huge knowledge-base. An example rule which tags the company name in the phrase *president of Digital Equipment* may be as follows:

Pattern: job position + "of" + capitalized word sequence
Action: Tag match as ENTITY, excluding job position and "of"

Although they outperformed all other sites participating in MUC-6 with an F-measure of 96.42%, and performed 98.27% in the training data, they could not repeat the same success in MUC-7. For name tagging, their performance was 91.32%. For this drop, they blame the MUC committee. They note that the domain of the formal test documents was different from the training and dry run documents. Although the NE task is defined as a domain-independent task, the MUC commiteee selected samples of New York Times articles that focused on particular domains, in the formal test. They think that this selection process greatly influenced the results of the task and substantially diminished the value of the formal test.

FASTUS

Perhaps the most famous rule-based information extraction system is FASTUS, a slightly permuted acronym for Finite State Automaton Text Understanding System, developed by SRI International, Artificial Intelligence Center [Hobbs *et al.*, 1996]. They note that it is an information extraction system, rather than a text understanding system, as its name implies. FASTUS is a set of cascaded non-deterministic finite-state automata, hence it is very fast. For the named entity extraction task, the first four levels of transducers are used:

1. *Tokenizer:* As its name implies, tokenizes the input text.
2. *Multi-word Analyzer:* Captures the collocations, like "in spite of".
3. *Preprocessor:* This is an optional phase, in which the developer can insert a transducer to handle more complex or productive multi-word constructs, like converting "twenty three" into a single number flag.
4. *Name Recognizer:* Performs the named entity extraction task.

The rules of the system were developed using a rule specification language, called FASTSPEC, that allows the developer to write regular productions, that are translated automatically into finite state machines by an optimizing compiler.

FASTUS did not participate in the recent MUC-7 evaluations, though they were one of the best perfomed sites in MUC-6, with an F-measure of 94% in the named entity extraction task.

LOLITA

The University of Durham participated in MUC-7 evaluations with the LOLITA System [Garigliano *et al.*, 1998]. This system was designed to be general purpose NLP system, and thus, named entity extraction, and even information extraction is only subset of this system. A core platform provides analysis and generation of natural language text. Upon this core, it is possible to develop information extraction, machine translation, natural language querying system, dialogue management, and a Chinese language tutoring systems. Perhaps, this is why their performance is this low (74.53%), they preferred to have a system capable of doing lots of things, instead of working well just for one task. The system has a core Semantic Network knowledge-base, and supported by the morphological analyzer, parser, semantic parser, and other tools.

Others

University of Sheffield, and Oki Electric Industry converted the named entity tagging task to a simple pattern matching problem [Humphreys *et al.*, 1998; Fukumoto *et al.*, 1998]. Using lists for persons, location, and organizations, they tried to tag words. Their performances were similar, and both obtained F-measures in low 80s.

7.3.2 Machine Learning Approaches

An alternative to rule-based systems is machine learning approaches, which are generally based on statistics. The systems in this category either used an n-gram language models (BBN's IdentiFinder, Kent Ridge Digital Labs, MITRE)

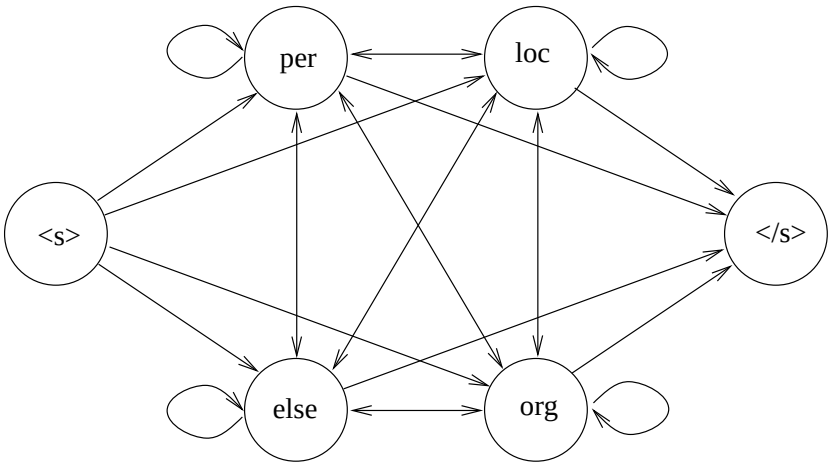

Figure 7.2: The conceptual structure of the basic HMM used by BBN for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, `per` denotes person, `loc` denotes location, `org` denotes organization, and `else` denotes that it does not belong to any of these categories.

or maximum entropy models (New York University's MENE) models.

IdentiFinder

BBN participated in MUC-7 with the IdentiFinder System [Bikel *et al.*, 1999; Miller *et al.*, 1998]. IdentiFinder is a hidden Markov model, that learns to recognize and classify name classes (names, dates, times, and numerical quantities, etc.). The conceptual structure of this HMM is depicted in Figure 7.2. They trained a bigram language model to get the most probable tag sequence from this HMM. The state observation probabilities were set to 1 for all states. Formally, they try to find the name class sequence maximizing $P(NC|W)$ in the text:

$$\underset{NC}{\operatorname{argmax}}\, P(NC|W)$$

Simplifying, using a bigram model one gets:

$$P(NC_i|W_i) \approx P_i(NC_i|NC_{i-1}W_{i-1}) \times P_i(W_i|NC_i, NC_{i-1})$$

However, using this model, it was impossible to distinguish whether there are two consecutive names, or is it a two-word name. So they decided to incorporate a special flag "+end+" so that the probability may be computed for any current word to be the final word of its name class. So the original formula becomes:

$$P(NC_i|W_i) \approx P_i(NC_i|NC_{i-1}W_{i-1}) \times P_i(W_i|NC_i, NC_{i-1}) \times P_i(+end+|W_i, NC_i)$$

In order to illustrate this formula, consider following sentence:

"Mr. Jones eats."

where "Jones" must be marked as "PERSON". The model would assign the following likelihood to this sequence:

$P(else| < s >,"+end+") \times$
$P(\text{“}Mr."|else, < s >) \times$
$P(+end+|"Mr", else) \times$
$P(per|else,"Mr.") \times$
$P(\text{“}Jones"|per, else) \times$
$P(+end+|"Jones", per) \times$
$P(else|per,"Jones") \times$
$P(\text{“}eats"|else, per) \times$
$P(+end+|"eats", else)$

Furthermore, they decided to augment the word information with word classes, such as `allCaps` (e.g. "KRDL"), `initCapNotCommonWord` (e.g. "David"), contains `DigitAndColon` (e.g. 2:34), etc. Now, what they called a word is a pair of words and word classes, and substituting W in the above formula with $< W, F >$ gives the final formula of IdentiFinder.

To classify the unknown words, they built another language model by replacing the unknown words with the flag "`_UNK_`", and if they encounter any unknown word in the test set, they used that language model.

IdentiFinder was the best performing statistical-only system in MUC-7 with an F-measure of 90.61%. Furthermore, their performance in the dry-run was much higher: 94%.

The Kent Ridge Digital Labs System

The Kent Ridge Digital Labs System [Yu *et al.*, 1998] is very similar to BBN's, but instead of using words in the language modeling, they used syntactic word classes similar to BBN's IdentiFinder. They augmented these classes with a knowledge-base containing stopwords, person, location, and organization lists, and other closed class word groups, such as names of the week days, moths, etc. Their performance was not so brilliant in MUC-7. They reached an F-measure of 76.65%, although their system can be used as an extension of BBN's system and maximum extropy models. They also used this system in Chinese name tagging.

MITRE's System

MITRE has also used a very similar approach to BBN's [Bikel *et al.*, 1999]. The only difference is that in order to prevent data sparseness, they used a class-based smoothing technique. Their main focus was on named entity extraction from spoken data.

MENE

New York University (NYU) participated in MUC-7 with a new system called MENE (Maximum Entropy Named Entity) [Borthwick *et al.*, 1998a; Borthwick *et al.*, 1998b]. Similar to any maximum entropy-based system, they use features, which can be classified in 5 types:

- *Binary Features:* These include binary valued features, such as "the token begins with a capitalized letter".

- *Lexical Features:* To create lexical history, such features are used. For example, if the previous word if "Mr.", then the current word is person". These features form the basis of their model.
- *Section Features:* These features make predictions based on the current section of the article, like "Date", "Preamble", and "Text".
- *Dictionary Features:* They automatically formed lists of names, and used them as additional data.
- *External Systems Features:* These set of features allow combining MENE with other systems. Features in this category checks whether other tagger predicts a word as a name or not.

MENE got an F-measure of 91% in the dry-run, but since test data was very different than training data, their performance decreased to 89% in official evaluations.

One superiority of the maximum entropy approach is that, it is easy to combine this system with other systems, and in that case, it is possible to reach an F-measure of 92% in the test set. In fact the HCRC-LTG system, described in the next section, is the proof of this hypothesis.

Alembic

In MUC-6 evaluations, MITRE participated with the Alembic system [Aberdeen *et al.*, 1995], using transformation-based error-driven learning algorithm of Brill [1993]. Their performance was in middle 80s.

RoboTag

Bennett *et al.* [1997] used binary decision trees using C4.5 [Quinlan, 1986] for name tagging task in the RoboTag system. The decision tree decides whether it is a name boundary or not. They use features indicating semantic properties

(like first name, title, corporate designator), locations (like city, country), part-of-speech tags (like adjective, noun), and token types (like upper/lower case). Their performance was 88.1% in the MUC-6 evaluation set.

Cucerzan and Yarowsky's System

An independent study by Cucerzan and Yarowsky [1999] attempts to build a language independent name tagger using a boot-strapping algorithm based on iterative learning. Re-estimation of contextual (e.g. "Mr.", "mayor of") and word-internal (e.g. "-oğlu" is a typical surname indicator in Turkish) patterns are captured in hierarchically smoothed trie models. This algorithm was evaluated for Romanian, English, Greek, Turkish, and Hindi. For Romanian, using a training set of 12,320 words, they reached an F-measure of 70.47%. For Turkish, training set was 5,207 words, and the final F-measure was 53.04%. This work was important in the sense that it was the first attempt for name tagging of Turkish.

7.3.3 Hybrid Approaches

Similar to other tasks in natural language processing, it is possible to get superior results by combining rule-based and statistical systems. The systems of University of Edinburgh and University of Manitoba are such examples in name tagging task.

LTG

The HCRC Language Technology Group (LTG) from the University of Edinburgh had an outstanding performance in recent MUC-7 evaluations [Mikheev *et al.*, 1998]. Their system has the best performance in name tagging with an F-measure of above 94%.

LTG is a hybrid system, and works in 5 phases, where first and third phases are rule-based, other phases rely on a pre-trained maximum entropy model.

1. *Sure-fire Rules:* In this phase, the systems marks names if there is enough surrounding context to decide. Sure-fire rules rely on only known corporate designators (like "Ltd.", "Inc."), titles (like "Mr.", "Dr."), and other similar definite contexts (like "shares of ORG").

2. *Partial Match 1:* In this phase, the systems performs a probabilistic partial match of the entities indentified in the document. First it enumerates the candidate names, which are substrings of the names found in the first phase. For example, if "Lockheed Martin Production" was tagged in the first phase, all instances of "Lockheed Martin Production", "Lockheed Martin", "Lockheed", "Martin", etc. were candidate names. These candidates were then evaluated using a pre-trained maximum entropy model. It takes into account contextual information for named entities, such as their position in the sentence, whether these words exits in lowercase, and if they were used in lowercase in the document, etc. If the model gives enough probability for a candidate, it is marked as a name.

3. *Relaxed Rules:* This phase is similar to the first phase, but this time, the rules have much more relaxed contextual constraints.

4. *Partial Match 2:* In this phase the system performs another partial match to annotate names similar to Phase 2.

5. *Title Assignment:* Upto this point, titles of news wires, which are all capitalized was ignored. In this phase they were annotated.

The LTG system had certain advantages. First of all, successfully combining both rule-based and statistical approaches, they outperformed all of the sites in MUC-7 evaluations. Unlike other sites, their system is not dependent on the training data, and even though the test data is very different than the training data, as in MUC-7, they performed very well. Indeed, they explain this as follows: Their system does not even require lots of training data [Mikheev *et al.*, 1999]. Without any training data, using only internal evidence (e.g. indicators such as "Mr." for people or "Ltd." for organizations) as well as external evidence (contexts such as "XXX the chairman of YYY" as evidence that XXX is a person,

and YYY an organization), they can still obtain satisfactory results in name tagging. They could tag organizations with an F-measure of 85%, persons with an F-measure of 92%, and locations with an F-measure of 53%. If they feed the system 200 names of countries and continents, the F-measure for locations increase to 88%.

University of Manitoba's System

Another hybrid, though not that successful system was developed by University of Manitoba of Canada for MUC-7 [Lin, 1998]. They augment the manually coded pattern rules with the rules, extracted from the training data. Then they use contextual cues to tag the unknown words using a Naive-Bayes classifier. They use a collocation database to automatically extract rules. The entries of this database have the form (word, relation, relative). For example, from the phrase "brown dog" they extract the collocation (brown, A:r-jnab:N, dog), which means that the word "brown" is an adjective, which modifies its relative "dog". They do not check for long distance relationships, since they do not use a parser. They reached an F-measure of 86.07% in MUC-7. They note that if they did not use the unknown word classifier, this decreases to high 70s.

7.4 Motivation

For this task, a corresponding Turkish example is provided in Figure 7.3.

Note that, the morphemes added after the names are not considered to be a part of the name, in order to be consistent with its definition for English. In English there are only a few such cases, (such as "Fred's"), but since Turkish is a highly agglutinative language, theoretically there are infinite number of word formations as described in Chapter 3.

Because of this reason, sometimes it is very problematic to determine the extent of the name in Turkish. It is very ambiguous whether it is necessary to

```
...<ENAMEX TYPE="ORGANIZATION''>PKK</ENAMEX> lideri
<ENAMEX TYPE=''PERSON''>Abdullah Öcalan</ENAMEX>, <ENAMEX
TYPE="LOCATION''>Yunanistan</ENAMEX>'da depreme yol açtı.
<ENAMEX TYPE="LOCATION''>Türkiye</ENAMEX>'ye karşı keskin
tutumuyla bilinen ve <ENAMEX TYPE="PERSON''>Apo</ENAMEX>'nun
<ENAMEX TYPE="LOCATION''>Kenya</ENAMEX>'ya götürülmesi konu-
sunda başrol oynadığı iddia edilen Dışişleri Bakanı <ENAMEX
TYPE="PERSON''>Teodoros Pangalos</ENAMEX> ile birlikte İçişleri
Bakanı <ENAMEX TYPE="PERSON''>Alekos Papadopulos</ENAMEX>
ve Kamu Düzeni Bakanı <ENAMEX TYPE="PERSON''> Filipos
Peçalnikos</ENAMEX> istifa ettiler....
```

Figure 7.3: An example of an example Turkish news article, whose named entities are marked.

begin with a capital letter and use an apostrophe in the location modifier. The Turkish Offical Grammar Guide [1996] is also insufficient to resolve this ambiguity. According to this guidebook, it is correct to write *Beyşehir Gölü*, but incorrect to write *Balkaş Gölü*, since *Balkaş* is not a name of a town unlike *Beyşehir*. So you have to know the names of all towns in Turkey, in order to capitalize the common nouns.

```
''Gediz Nehri'nde''
<ENAMEX TYPE="LOCATION">Gediz Nehri</ENAMEX>'nde

''Marmara denizinde''
<ENAMEX TYPE="LOCATION">Marmara</ENAMEX> denizinde
```

This characteristic of Turkish also effects other types of names, too.

```
''Ahmetler'e gidiyoruz.''  (We are going to Ahmet's)
<ENAMEX TYPE=''PERSON''>Ahmetler</ENAMEX>'e
```

The agglutinative nature of Turkish also impacts negatively the use of lexical modeling based on only words. Similar to topic segmentation task, data sparseness is also a problem in name tagging. This is why we separated the tokens with apostrophe into two parts. Such an approach prevented our model suffer from data sparseness at least for the named entities. Hence we can expect the damage

of the data sparseness to be more destrcutive when the input is audio data, where there is no apostrophe punctuation after the named entities. A detailed analysis of such cases are provided in Section 7.8.5.

Thus, we have to augment the lexical model with contextual and morphological models. Next section describes our approach and models in detail.

7.5 Approach

Our approach is based on n-gram language models embedded in hidden Markov models. We used the following four models in the name tagging task:

- *Lexical Model*, which captures the lexical information using only word tokens.
- *Contextual Model*, which captures the contextual information using the surrounding context of the word tokens. This model is especially helpful in tagging unknown words.
- *Morphological Model*, which captures the morphological information with respect to the corresponding case and name tag information. In order to build this model, we used the morphological parses of the words.
- *Name Tag Model*, which captures the name tag information (person, location, organization, and else) of the word tokens.

Each model is smoothed using Good-Turing method [Good, 1953] combined with the Back-off modeling proposed by Katz [1997], as described in Chapter 2. In this work, in order to build a language model, and decode the most probable output in an HMM with the Viterbi algorithm [Viterbi, 1967], we used the publicly available SRILM toolkit, developed by Andreas Stolcke [Stolcke, 1999]. We would like to explain each model in detail in the following subsections, then discuss on the methods for combining these four models.

7.5.1 Lexical Model

For lexical modeling, we used a simplified version of BBN's name finder [Bikel *et al.*, 1999]. The states of the hidden Markov model were word/tag combinations, where the tag indicated whether a word was part of a proper name, and of what type (person, place, or organization). Transition probabilities consisted of trigram probabilities over these combined tokens. The word/tag observation likelihoods for each state was set to 1.

In order to detect the boundaries of the names, we used a fictitious boundary flag. This flag holds one the following three values:

1. *yes:* indicates that there is a name boundary.
2. *no:* indicates that there is no name boundary.
3. *mid:* indicates that the previous and the next tokens belong to the same name.

The conceptual structure of this HMM is depicted in Figure 7.4. Note that, although it is possible to get a sequence of "person mid organization", the use of language model discourages such transitions for all cases. This is why we did not need to put a separate "mid" boundary state for each these 3 name types.

An example will clarify this notation. Consider following piece of annotated text:

```
<ENAMEX TYPE="ORGANIZATION">Bilkent University</ENAMEX>'s <ENAMEX
TYPE="ORGANIZATION">Graduate School of Business</ENAMEX> is in Ankara.
```

The corresponding output sequence for this text would be as follows:

"<s> boundary/*yes* Bilkent/*organization* boundary/*mid* University/*organization* boundary/*yes* 's/*else* boundary/*yes* Graduate/*organization* boundary/*mid* School/*organization* boundary/*mid* of/*organization* boundary/*mid* Business/*organization* boundary/*yes* is/*else* boundary/*no* in boundary/*yes* Ankara/*location* boundary/*yes*

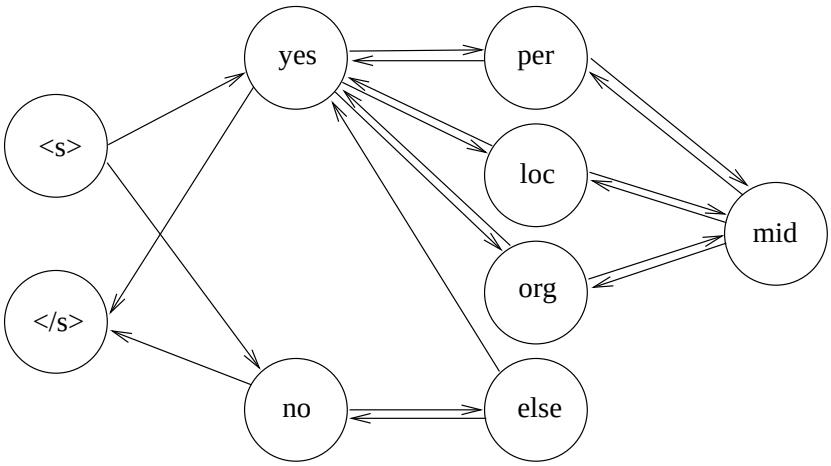

Figure 7.4: The conceptual structure of the basic HMM for name tagging. <s> denotes the start of sentence, and </s> denotes the end of sentence, `yes` denotes the name boundary, `no` denotes that there is no name boundary, `mid` denotes that it is in the middle of a name, `per` denotes person, `loc` denotes location, `org` denotes organization, and `else` denotes that it does not belong to any of these categories.

</s>"

This implies that, name tagging task does not only require tagging each word with one of the 4 possible tags (person, location, organization, and else), but also detecting the boundaries. In fact, using this boundary flag also improved the tagging performance. This flag has also performed as a connection between the surrounding tokens. Consider the following example:

```
<ENAMEX TYPE="ORGANIZATION">Ankara Üniversitesi</ENAMEX>
```

The city "Ankara" can either be location or a part of an organization. As seen from the Table 7.2, the boundary flag helps us to find the correct tagging, since the trigram "Ankara/*organization* boundary/*mid* Üniversitesi/*organization*" is about 4000 times more probable than the trigram "Ankara/*location* boundary/*yes* Üniversitesi/*organization*", although tagging "Ankara" as location is more probable. The reason for this difference is that there is no occurance of the bigram "boundary/*yes* Üniversitesi/*organization*", but lots of "boundary/*mid* Üniversitesi/*organization*". This is why marking the whole phrase as a location is more probable than separating them.

Output Sequence	Probability
Ankara/*organization* boundary/*mid* Üniversitesi/*organization*	0.015029
Ankara/*location* boundary/*yes* Üniversitesi/*organization*	0.000004

Table 7.2: The effect of the boundary flag on the performance of the tagger.

Output Sequence	Probability
Dr./*else* boundary/*yes* unk/*person*	0.990119
Dr./*else* boundary/*yes* unk/*location*	0.000690
Dr./*else* boundary/*yes* unk/*organization*	0.000880
Dr./*else* boundary/*else* unk/*else*	0.002688

Table 7.3: The use of the contextual model for unknown words.

7.5.2 Contextual Model

For contextual modeling, we improved our lexical language model as follows: We marked as unknown every other word in our training data, and then built a language model, then interpolated this model with the lexical model. Using this contextual model, we could tag the unknown words by looking at the context. This idea has first been used in [Hakkani-Tür *et al.*, 1999]. For example, the word after the abbreviation "Dr." is generally a person, The word "University" is often a part of an organization.

In order to demonstrate this model with a real example, consider this piece of text:

```
Dr.  <ENAMEX TYPE="PERSON">Tür</ENAMEX>
```

Assuming that the word "Tür" is unknown, i.e. did not appear in the training data, we can use the contextual model to tag this word by replacing it with the flag "unk", and let the model choose for the maximum probable tag considering the neighboring word "Dr.". Table 7.3 gives the probabilities of the output sequences in which "Tür" is tagged as person, location, organization, or else, assuming that "Dr." is not a part of the name.

More formally, this model helps tagging unknown words by modeling the following 4 clues:

1. Previous token in the same name, e.g. First names of the persons in a context like "Gökhan Tür", assuming that first names are a smaller set than the surnames,
2. Previous token outside of the name, e.g. "Mr.", "Dr.", "Sayın", in a context like "Sayın Tür",
3. Next token in the same name, e.g. "Üniversitesi", "Hastanesi", in a context like "Manitoba Üniversitesi",
4. Next token outside of the name, e.g. "'de", "kentinde", in a context like "İzmir'de", or "İzmir kentinde".

These cues can be considered as the help of prepositions in English. Since, Turkish is an agglutinative language, there are no prepositions, but corresponding suffixes are attached to words. If the word is a proper name, the word and the suffix are separated using an apostrophe. We considered these suffixes after the apostrophe as separate tokens, and this helped us a lot in contextual modeling.

7.5.3 Morphological Model

For morphological modeling, we morphologically analyzed all the words in our training data using the analyzer developed by Oflazer [1993], disambiguated them using the statistical morphological disambiguator of Hakkani-Tür [2000], and used the morphological parses of the words while training, instead of the surface forms. See Chapter 3 for a detailed discussion of Turkish morphological structure.

We also added case information to the morphological parses, to indicate whether:

- the word is all in lower case, (NOCAP), e.g. "ev" (house),

- the word is all in upper case, (ALLCAP), e.g. "CNN", or
- only the initial letter of the word is in upper case, (CAP), e.g. "Demirel". For this case, we did not mark whether it is sentence initial or not.

We expected the morphological analyses of the words would help us in two ways:

1. Our morphological analyzer has a proper name database, and marks common Turkish person, location, and organization names as proper. In the morphological model, we can expect words, marked as proper are also to be marked as names.
2. Besides this, the names are mostly noun phrases, and during training, we can expect the morphological model to learn such patterns. For example consecutive two proper nouns is a common person pattern, as in "George Washington".

Since the lexicon of our morphological analyzer does not distinguish proper nouns with respect to their types, and there is no other way for this model to distinguish different names syntactically, morphological model only decides whether a word is a name or not. While tagging using only morphological model, we tag the words marked as name with the most popular name type, i.e. "person". While combining this model with other models, we give the same probability to all of the name types.

Let us demonstrate these expectations using a concrete example. Similar to Tables 7.2 and 7.3, Table 7.4 gives the probabilities for the named entity:

<ENAMEX TYPE="PERSON">Süleyman Demirel</ENAMEX>

where, both "Süleyman" and "Demirel" are analyzed as:

"Noun+Prop+A3sg+Pnon+Nom+CAP".[3]

[3]See Appendix A for the definition of features in this morphological parse.

Output Sequence	Probability
Noun+Prop+A3sg+Pnon+Nom+CAP/*person* boundary/*mid* Noun+Prop+A3sg+Pnon+Nom+CAP/*person*	0.300339
Noun+Prop+A3sg+Pnon+Nom+CAP/*else* boundary/*no* Noun+Prop+A3sg+Pnon+Nom+CAP/*else*	0.0231911

Table 7.4: The use of the morphological model.

7.6 Tag Model

The tag model is a trigram language model, which does not include any lexical items, but only the name tags, i.e. person, location, organization, and else, and the boundary flag types, i.e., yes, no, and mid. So its vocabulary consists of these 7 tokens. We built it by extracting the lexical words in our training data, and leaving only these tags.

The idea of developing a tag model was suggested by the result of the analysis of the errors of our name tagger. We found out that, some multi-token names were separated into different names of same or different types. For example the name

```
<ENAMEX TYPE="PERSON">Alaattin Eroğlu</ENAMEX>
```

was incorrectly tagged as

```
<ENAMEX TYPE="PERSON">Alaattin</ENAMEX>
<ENAMEX TYPE="PERSON">Eroğlu</ENAMEX>
```

Such a tagging damages the performance in two ways:

1. One of the names is marked as "spurious" by the evaluation software.
2. The other one's *type* is correct, but *text* is marked as wrong[4].

[4]See Section 7.8.2 for a detailed explanation of the evaluation metrics.

Output Sequence	Probability
person mid person	0.999870
person yes person	0.006076

Table 7.5: The use of the tag model.

On the other hand, the tag models favors for the correct tagging as seen in Table 7.5.

In other words, the function of this model is to limit the improbable tag sequences, rather than finding names. Thus, we can expect the number of spurious and incomplete tags in our output to decrease, hence our performance to increase.

7.7 Model Combination

It is possible to tag a text using the lexical model or the morphological model alone. This is not the case for other two models. Since morphological model does not include any lexical information, we do not expect the performance of the tagger to be high using only this model.

In order to tag using only lexical model, we set the state observation likelihoods to 1, and use only the lexical model in Viterbi decoding[5]. Similarly, in order to tag using the morphological model, we first convert the tokens into their morphological parses, and use Viterbi decoding, then reconvert them into their original forms.

In order to combine the lexical model with the contextual model, we simply weighted interpolated these two models. The optimum weight is chosen using a separate held-out set. This mixture model can then be used in Viterbi decoding.

[5]See Chapter 2 for a detailed description of these concepts.

Combining lexical model and the morphological model is not that easy. Instead of interpolating the models, we have to interpolate the posterior probabilities, since one uses lexical forms of the words, while the other uses the morphological parses. We interpolated the posterior probabilities using empirically optimized weighting using a separate held-out set. After this interpolation, we can select the most probable tag for each word.

More formally, using lexical model, we can compute:

$$P_{LM}(w_i/t_i|w_{i-2}/t_{i-2}, w_{i-1}/t_{i-1})$$

where LM denotes lexical model, w_i denotes the i^{th} word (this can be either a real word, or a boundary), and t_i denotes the tag of that word.

Using our HMM, we can also compute the posterior

$$\sum_{t_{i-1},t_{i-2}} P_{LM}(w_i/t_i|w_{i-2}/t_{i-2}, w_{i-1}/t_{i-1}) = P_{LM}(w_i/t_i|w_{i-2}, w_{i-1})$$

$P(w_i/t_i) = P(t_i|w_i)$, since w_i is given. Hence, we can rewrite the above formula as follows:

$$P_{LM}(t_i|w_{i-2}, w_{i-1}, w_i)$$

Similar to this notation, the morphological model can give us the posterior:

$$P_{MM}(M(w_i)/t_i|M(w_{i-2})/t_{i-2}, M(w_{i-1})/t_{i-1})$$

where MM denotes morphological model, $M(w)$ denotes the morphological analyses of the word w. Following the above notation we can say that this posterior is equal to:

$$P_{MM}(t_i|M(w_{i-2}), M(w_{i-1}), M(w_i))$$

Then, we can simply interpolate these posteriors with some weight λ. A top level representation of this interpolation can be written as follows:

$$P_{LM+MM}(T|W, M(W)) = \lambda P_{LM}(T|W) + (1 - \lambda) P_{MM}(T|M(W))$$

where T denotes the sequence of tags, t_i, W denotes the input string, $M(W)$ denotes the morphological analyses of the words in the input string, $M(w_i)$.

Combining the morphological model with the mixture of the lexical and the contextual models can also be possible by interpolating the posterior probabilities obtained these information sources. The formal equations for this combination are very similar to combining morphological and lexical models, hence left as an exercise.

Up to this point the tag model is not used in the combinations. In fact, the use of the tag model needs a little trick. In order to use this model, we used the posterior probabilities obtained from any combination of the other three models as state observation likelihoods, and use the tag model in order to determine the transaction probabilities. One problem with this operation is converting posteriors, $P(T|W)$, to likelihoods, $P(W|T)$. This conversion is possible using the Bayes' rule:

$$P(W|T) = \frac{P(T|W)P(W)}{P(T)}$$

Since we use try to optimize the output sequence, and $P(W)$ is given, hence constant, division of the posteriors to priors is proportional to the likelihood, and can be used in Viterbi decoding. In this HMM, the transition probabilities can be obtained using the tag model.

Combining all models can be stated more formally as follows:

$$P_{LM+MM+CM+TM}(T|W, C(W), M(W), T(W)) \propto \frac{P_{LM+CM+MM}(T|W,C(W),M(W))}{P(T)} \times P_{TM}(T)$$

where CM denotes contextual model using contexts of the words, $C(W)$, TM denotes tag model using the tag sequence $T(W)$, λ is an emprically determined balancing parameter to adjust the dynamic ranges of the combined models.

Figure 7.5 shows a set of possible combinations of four models. Note that, there are also other ways of combining these models. For example, it is possible to combine lexical and tag models, by obtaining the posteriors from the lexical model, convert to likelihoods, and decode using the tag model as transition probabilities.

7.8 Experiments and Results

In this section, we report the results of evaluating the Turkish name tagger using the MUC evaluation software. In order to better understand the power of the models, and their combinations, we also present results for tagging English, using same models and evaluation metrics.

7.8.1 Training and Test Data

We trained our system using 492,821 words of newspaper articles containing 16,335 person names, 11,743 location names, and 9,199 organization names, summing up to 37,277 names. For testing we used about 28,000 words of newspaper articles, containing 924 person names, 696 location names, and 577 organization names, summing up to 2,197 names.

7.8.2 Evaluation Metrics

Along with the definition of the named entity extraction task, the evaluation metrics are also set by the MUC program. MUC scoring software is used to evaluate the systems participated in these conferences.

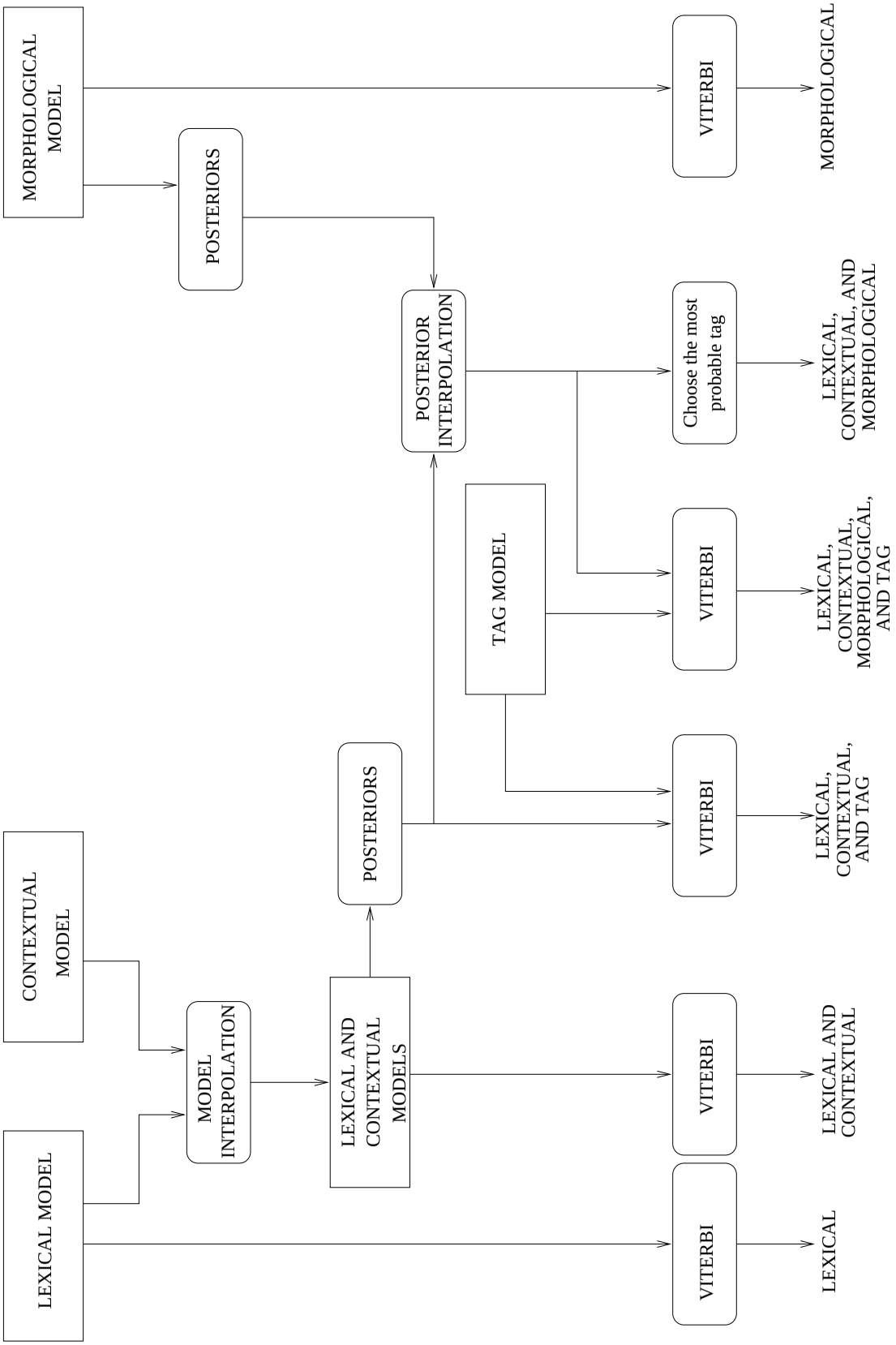

Figure 7.5: Combining lexical, contextual, morphological, and tag models for tagging Turkish text.

For the name tagging task, there are 2 criteria to evaluate:

- **Type:** Checks for the type of the name, i.e. person, location, or organization.
- **Text:** Checks for the text marked as a name.

For each of these 2 criteria, the evaluation software computes the following 3 values:

- **Correct:** Number of correct answers found by the name-finder.
- **Actual:** Number of answers found by the name-finder.
- **Possible:** Number of possible correct answers in the key.

For *Type* and *Text* criteria, the above 3 values are summed up. Then, borrowed from the information retrieval community, *recall* and *precision* values are computed as follows:

$$Recall = \frac{Correct\ Type + Correct\ Text}{Possible\ Type + Possible\ Text}$$

$$Precision = \frac{Correct\ Type + Correct\ Text}{Actual\ Type + Actual\ Text}$$

Informally, recall measures the number of hits vs. the number of possible correct answers as specified in the key, whereas precision measures how many answers were correct ones compared to the number of answers delivered. There is no partial credit in *Text* criterion. Even though most of the words of a name have been marked, this is called as incorrect.

Finally, these two measures of performance are combined to form one measure of performance, the *F-measure*, which is computed by the uniformly weighted harmonic mean of precision and recall:

	Possible	*Actual*	*Correct*
Text	4058	4048	3993
Type	4058	4051	3980
Total	8116	8099	7973
F-Measure	98.34%		

Table 7.6: An example output of the MUC scorer.

Model	*Text*	*Type*	*F-Measure)*
Lexical	80.87%	91.15%	86.01%
Morphological	36.52%	79.73%	58.12%
Lexical+Contextual	86.00%	91.72%	88.86%
Lexical+Contextual+Morphological	87.12%	92.20%	89.66%
Lexical+Contextual+Tag	89.54%	92.13%	90.84%
Lexical+Contextual+Morphological+Tag	**90.40%**	**92.73%**	**91.56%**

Table 7.7: Accuracy of the name tagging task using lexical, contextual, morphological, and tag models.

$$F - Measure = \frac{Recall \times Precision}{\frac{1}{2} \times (Recall + Precision)}$$

Table 7.6 demonstrates these metrics.

7.8.3 Results

Table 7.7 gives the accuracy of our system according to the MUC evaluation metrics. We have provided results using only lexical and morphological information in addition to the four types of combinations shown in the table, although it is possible to combine these information sources in eleven different ways. In all of the combinations, we did not separate the lexical model from the contextual model, because lexical model alone is relatively very weak in tagging. So we are left with only four types of combinations.

	Possible	*Actual*	*Correct*	*F-Measure*
Person	927	945	867	92.63%
Location	698	716	674	95.33%
Organization	576	607	531	89.77%
TOTAL	2201	2268	2072	92.73%

Table 7.8: Detailed name tagging results.

We are very pleased to see that, lexical model alone performed high 80s. When we look at this model in detail, we see that we have done well in detecting the types of the names, but we have problems in detecting them. The main reason of this problem is the unknown words. This problem is solved by the contextual model, and the performance of the "Text" metric is increased to 86%. It is also interesting to see that the morphological model alone has performed about 58%, without even knowing the surface forms or the roots of the words, a score which was not expected even by us. We were also successful in incorporating the extra information held by the morphological model to the combination of lexical and contextual models, and gained 0.8% points more. Instead of the morphological model, when we have incorporated the tag model, we have gained about 2% points more. These improvements are important, since we have entered a range, in which it is very hard to achieve further improvements. Finally, when we have combined all of our models, we have reached 91.56%. We see that tag model is very effective in this task. The "Text" metric is increased more than 3% points, and "Type" metric is increased about 0.5% points in either cases this model was used. Similarly, the morphological model increases the "F-Measure" by 0.8% in either cases it was used. When we compare the final "F-Measure" with our baseline lexical performance, we see an improvement of 5.55% points.

7.8.4 Error Analysis

Table 7.8 shows the performance of our name tagger with respect to name types. These are the results when we use all four of our models.

We see that our performance varies greatly for different name types. It is also interesting to see that, our performance is best for locations, and worst for organizations. When we analyze our test data we see that our system performs not so satisfactory for very long organization names. For example the organization:

<ENAMEX TYPE=''ORGANIZATION''>Adana Emniyet Müdürlüğü Organize Suç ve Silah Kaçakçılığı Şube Müdürlüğü''</ENAMEX>

was tagged as:

<ENAMEX TYPE=''ORGANIZATION''>Adana Emniyet Müdürlüğü Organize Suç ve Silah</ENAMEX> Kaçakçılığı <ENAMEX TYPE=''ORGANIZATION''>Şube Müdürlüğü''</

which results in two different names, neither of which were tagged as correct in "Text", and only one was tagged as correct in "Type".

7.8.5 Effect of the Case and Punctuation Information

Tagging Turkish becomes more critical when we remove case and punctuation information. Such an experiment is important in order to see the performance of the tagger with speech recognizer output (SNOR) format, which is largely unpunctuated (apostrophes are preserved) and in all capital letters, as set by the NIST [1998]. Case is a very valuable information source in tagging proper names. Similarly punctuation has importance for this task in detecting the name boundaries, since most of the time, punctuation resolves ambiguities, such as "Mesut, Yılmaz" vs. "Mesut Yılmaz". While deleting the punctuation, we did not touch the apostrophe sign, since only this punctuation is provided with the speech recognizer output. This nuance has extra importance for tagging Turkish, because we have been modeling a proper name and its suffixes separately. This is easy, because the apostrophe sign marks the boundary between the root and the suffixes. If we had to remove all the punctuations, we would lose this information, and we would expect to face with data sparseness in building our model.

These experiments are necessary because of the following reasons:

Model	*Text*	*Type*	*F-Measure)*
Lexical	80.71%	90.17%	85.44%
Lexical+Contextual	84.26%	90.72%	87.49%
Lexical+Contextual+Tag	90.88%	85.28%	88.08%

Table 7.9: Accuracy of the name tagging task using lexical, contextual, and tag models without case and punctuation information.

- When we are going to tag proper names from speech input, there will be no case or punctuation. So we have to see how well we are doing for such input.
- We can now easily see the effectiveness of our method while tagging such input for an agglutinative language.

In these experiments, our aim is not to improve our performance with input lacking case or punctuation, but instead to see our performance without any modification to the models and system.

Table 7.9 shows our results when we drop the case and/or punctuation information. We see that our models can still be used in such a case since we did suffer too much. The decrease in the performance was 2.76% when using lexical, contextual, and tag models. Indeed, these results are comparable with the results of BBN [Bikel *et al.*, 1999]. They have reported a loss of 4.2% (from 94.9% to 90.7%) on the Wall Street Journal articles using the SNOR format.

7.8.6 Results Compared to Name Tagging of English

In order to see whether these results are comparable with the results obtained for English, we built a similar system using similar statistical methods. Table 7.10 presents the performance of our algorithm applied to both English and Turkish input in SNOR format.

Language	*Text*	*Type*	*F-Measure*
Turkish	84.26%	90.72%	87.49%
English	82.95%	89.56%	86.26%

Table 7.10: Comparison of the Turkish and English name tagging results using only lexical and contextual models.

7.9 Conclusion

We presented a probabilistic model for automatically tagging names in a Turkish text. We used four different information sources to model names, and successfully combined them. Our first information source is based on the surface forms of the words. Then we combined the contextual clues, and obtained a significant win. After this, we modeled the morphological analyses of the words, and reached an F-measure of 89.66% according to the MUC evaluation software, which was 3.65% points better than the lexical model alone. Finally, we modeled the tag sequence, and gained 1.90% more, reaching an F-measure of 91.56% in Turkish name tagging.

This was the second study on Turkish named entity extraction. Cucerzan and Yarowsky [1999] reported an F-measure of 53% using very little training data. This implies the importance of the size of the training data in corpus based language processing tasks. The huge difference in the training data sizes makes a comparison impossible. Instead, we gave results for English, using the same lexical and contextual models, and see that it is possible to reach an F-measure of 86%.

These results are important in the following senses:

- We have successfully combined lexical, contextual, morphological, and tag information for this basic information extraction task. Each model contributed to this task as we have expected.
- Using the lexical model alone performed high 80s for Turkish name tagging, which is a very similar result we obtained for English. Thus, we can

claim that statistical methods can be used for name tagging task even for agglutinative languages.

- We have seen that removing the case and punctuation (except the apostrophe sign) information results in a 2.76% points decrease in F-measure. This implies that our models can still be used in such a case .

- Recalling that named entity extraction task is a prerequisite task for other more complex information extraction tasks, we are now ready to move on other tasks. If Multilingual Entity Task (MET) conferences are generalized to handle more languages, we are ready to participate for Turkish, and other morphologically rich languages, given we have enough training data.

- As a future research, we would like to work on the problems we have encountered in tagging organizations.

Chapter 8

Conclusion

We have presented statistical solutions to various information extraction tasks for Turkish. We can list the tasks we have worked on as follows:

- The *Turkish Text Deasciifier* task aims to convert the ASCII characters in a Turkish text, into the corresponding non-ASCII Turkish characters (i.e., "ü", "ö", "ç", "ş", "ğ", "ı", and their upper cases).
- The *Word Segmentation* task aims to detect word boundaries, given we have a sequence of characters, without space or punctuation.
- The *Vowel Restoration* task aims to restore the vowels of an input stream, whose vowels are deleted.
- The *Sentence Segmentation* task aims to divide a stream of text or speech into grammatical sentences. Given a sequence of (written or spoken) words, the aim of sentence segmentation is to find the boundaries of the sentences.
- The *Topic Segmentation* task aims to divide a stream of text or speech into topically homogeneous blocks. Given a sequence of (written or spoken) words, the aim of topic segmentation is to find the boundaries where topics change.

Task	Information Used (Other Than Lexical)
Deasciifier	None
Word Segmentation	None
Vowel Restoration	None
Sentence Segmentation	Final IGs of the morphological analyses
Topic Segmentation	Stems of words and nouns
Name Tagging	Final IGs of the morphological analyses, Contextual and tag models

Table 8.1: Summary of the tasks, and the information sources other than the lexical information used in this thesis.

- The *Name Tagging* task aims to mark the names (persons, locations, and organizations) in a text.

Table 8.1 summarizes the tasks presented in this thesis, and the information sources used other than the words themselves. We can say that for simple tasks, only lexical information is enough, but in order to obtain better performance in more complex tasks, like topic segmentation and name tagging, we do not only use lexical information, but also morphological, and contextual information. For sentence segmentation, we have also modeled the final IGs of the words and combined it with the lexical model. For name tagging, in addition to the lexical and morphological models, we have also employed contextual and tag models. For topic segmentation stems of the words (or nouns) have been found to be more effective than using the surface forms of the words.

When we look at the training data size we have used, all tasks except the name tagging task, have been trained using a training corpus of 18 million words. Since we had to manually annotate the training data for the name tagging task, this number is only 500,000 words for name tagging. Because of the extra time consumed for annotation, name tagging task took much longer than the other tasks.

We can list the importance of these results as follows:

- Statistical methods have been largely ignored for processing Turkish. Mainly due to the agglutinative nature of Turkish words and the structure of Turkish sentences, the construction of a language model for Turkish can not be directly adapted from English. It is necessary to incorporate some other techniques. This work is a preliminary step in the application of corpus-based statistical methods to Turkish text processing.

- The most important engineering contribution of this thesis is the system that has been developed: An information extraction system for Turkish. In our view, regardless of the method and technologies used, developing such a system for the first time for Turkish is as important as developing an information retrieval, a speech recognition, or a machine translation system for Turkish.

- This study is also a pioneering work in Turkish text understanding, since sentence segmentation, topic segmentation, and name tagging tasks are the preliminary steps of more complex tasks on the way to text understanding.

The main problems in applying statistical methods to Turkish can be listed as follows:

- The lack of training data was the most important problem in this work. We have tried to compile a corpus of about 19 million words from the web resources of Milliyet newspaper. Still, our corpus problem has not been solved, because, first of all we had to clean-up the text, which was not an easy task, then for name tagging task, we manually annotated about 500,000 words of this corpus. The need for a large, clean, and annotated corpus is still valid for other tasks, such as morphological disambiguation, parsing, more complex IE tasks, etc.

- In our work, the agglutinative nature of Turkish was the second biggest problem. Note that we have mentioned two main characteristics of Turkish in Chapter 3. These were the free word order of the sentences and the agglutinative structure of the words. We did not suffer from the free word

order, because of the nature of the tasks we were dealing with. But if we had attempted to develop a probabilistic parser, for example, this would have been a big problem. We have proposed several solutions to the problems due to the agglutinative structure of the Turkish words.

- Another problem was the lack of studies in Turkish information extraction. We have had no chance in comparing our performance with other studies. Instead, for some tasks, we have built similar systems for English, and compare these results with other systems developed for English.

Finally, it is time to conclude this thesis with some future directions:

- After these basic information extraction tasks, we are now ready to improve these systems, and build more sophisticated systems using the same framework. For example BBN's IdentiFinder system [Bikel *et al.*, 1999] includes all information extraction systems in MUC conferences using a unique statistical framework.

- Scientifically, it is possible to investigate for more sophisticated solutions to data sparseness problem due to the agglutinative nature of Turkish. We have only employed word-based syntactic information, i.e. morphological anayses of the words, and ignored phrase or sentence-based syntactic information. This may help especially for sentence segmentation and name tagging. This is also an open research area in English, which was ignored due to its cost.

- It is possible to build hybrid systems, such as the named entity extraction system of Mikheev et al. [1998]. We have used only statistical models for all tasks. It is sometimes useful to combine rule-based systems with statistical ones.

- Especially for the name tagging task, there is still some room for improvement using more training data. This is also valid for other Turkish text processing tasks, such as parsing or morphological disambiguation.

- It would be great if we can combine these tasks in one stand- alone system with a user interface, like MAESTRO of SRI International [Rivlin *et al.*, 1999], or Alembic of MITRE [Aberdeen *et al.*, 1995].

- All the tasks presented in this thesis can also be used for the speech recognizer output. We have provided results using no case or punctuation information, but if we had a speech recognizer output data, we could try the performance on that noisy data, and maybe try the use of prosody for these tasks, since prosody has been proved to be very effective in especially segmentation tasks [Stolcke *et al.*, 1999; Tür *et al.*, 2000; Shriberg *et al.*, 2000; Hakkani-Tür *et al.*, 1999].

- It is possible to adopt the same techniques in order to build statistical systems for other agglutinative languages. Our intuition is that, it is possible to incorporate part-of-speech information into some of the systems built for English, such as topic and sentence segmentation, and name tagging. Recall that clustering only nouns instead of all words resulted in a huge win for the topic segmentation task.

Appendix A

Turkish Morphological Features

Feature	*Definition*
^DB	Derivation boundary
A3sg	Third person singular agreement
A3pl	Third person plural agreement
Abl	Ablative case
Acc	Accusative case
Adj	Adjective
Agt	Agent
Become	Become verb
Caus	Causative verb
Conj	Conjunctive
Det	Determiner
FitFor	FitFor
Gen	Genitive case
Ins	Instrumantive case
Loc	Locative case
Nom	Nominative case
Noun	Noun

`P1sg`	First person singular possessive agreement
`P2sg`	Second person singular possessive agreement
`P3sg`	Third person singular possessive agreement
`P3pl`	Third person plural possessive agreement
`PastPart`	Derived past participle
`Pnon`	No possessive agreement
`Pos`	Positive polarity
`Prop`	Proper name
`Verb`	Verb
`With`	With
`Without`	Without
`Zero`	Zero derivation with no overt derivation

Appendix B

Turkish Stopword List

ve	bir	bu	da	eğer
de	için	ile	olarak	kabul
daha	çok	en	gibi	artık
sonra	olan	ama	kadar	tam
ne	ise	iki	var	çünkü
büyük	yeni	her	o	bütün
ilk	son	ancak	değil	iş
dedi	diye	olduğunu	ki	mı
önce	yüzde	olduğu	göre	diğer
içinde	bin	yok	iyi	bunu
milyon	zaman	karşı	söyledi	milyar
arasında	ya	ilgili	gün	nasıl
oldu	tarafından	yer	kendi	başka
önemli	hem	bile	mi	eden
aynı	dün	ortaya	gelen	geçen
tek	böyle	biz	ben	yine
kez	etti	sadece	siyasi	şimdi
bunun	birlikte	özel	konusunda	şu
nedeniyle	hiç	şöyle	üzerine	eski
yaptığı	bugün	yapılan	devam	bazı
tüm	pek	eğer		

Bibliography

[Aberdeen *et al.*, 1995] Aberdeen, J.; Burger, J.; Day, D.; Hirschman, L.; Robinson, P.; and Vilain, M. 1995. MITRE: Description of the Alembic system used for MUC-6. In *Proceedings of the MUC-6.*

[Allan *et al.*, 1998] Allan, J.; Carbonell, J.; Doddington, G.; Yamron, J.; and Yang, Y. 1998. Topic Detection and Tracking Pilot Study: Final Report. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA. 194–218.

[Appelt and Israel, 1999] Appelt, D. E. and Israel, D. J. 1999. Introduction to Information Extraction Technology. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence.*

[Bahl *et al.*, 1983] Bahl, Lalit R.; Jelinek, Frederick; and Mercer, Robert L. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5(2):179–190.

[Beeferman *et al.*, 1999] Beeferman, Doug; Berger, Adam; and Lafferty, John 1999. Statistical Models for Text Segmentation. *Machine Learning* 34(1-3):177–210. Special Issue on Natural Language Learning.

[Bennett *et al.*, 1997] Bennett, S. W.; Aone, C.; and Lovell, C. 1997. Learning to Tag Multilingual Texts through Observation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.*

[Bikel *et al.*, 1999] Bikel, Daniel M.; Schwartz, Richard; and Weischedel, Ralph M. 1999. An Algorithm that Learns What's in a Name. *Machine Learning* 34:211–231. Special Issue on Natural Language Learning.

[Black *et al.*, 1998] Black, W.J.; Rinaldi, F.; and Mowatt, D. 1998. FACILE: Description of the NE System Used for MUC-7. In *Proceedings of the MUC-7.*

[Borthwick *et al.*, 1998a] Borthwick, A.; Sterling, J.; Agichtein, E.; and Grishman, R. 1998a. NYU: Description of the MENE Named Entity System as Used in MUC-7. In *Proceedings of the MUC-7.*

[Borthwick *et al.*, 1998b] Borthwick, A.; Sterling, J.; Agichtein, E.; and Grishman, R. 1998b. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In *Proceedings 6th Workshop on Very Large Corpora at the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*, Montreal, Canada.

[Brill, 1993] Brill, Eric 1993. *A Corpus-Based Approach to Language Learning.* Ph.D. Dissertation, Department of Computer Science, University of Pennsylvania.

[Brown *et al.*, 1992] Brown, Peter F.; Della Pietra, Stephen A.; Della Pietra, Vincent J.; Lai, Jenifer C.; and Mercer, Robert L. 1992. An Estimate of an Upper Bound for the Entropy of English. *Computational Linguistics* 19(1):31–40.

[Charniak, 1993] Charniak, Eugene 1993. *"Statistical Language Learning".* The MIT Press.

[Chelba, 2000] Chelba, Ciprian 2000. *Exploiting Syntactic Structure for Natural Language Modeling.* Ph.D. Dissertation, Department of Electrical and Computer Engineering, Johns Hopkins University.

[Chinchor and Robinson, 1998] Chinchor, N. and Robinson, P. 1998. MUC-7 Named Entity Task Definition (version 3.5). In *Proceedings of the MUC-7.*

[Chomsky, 1969] Chomsky, Noam 1969. Quine's Empirical Assumptions. In Davidson, Donald and Hintikka, Jaakko, editors 1969, *Words and objections. Essays on the work of W. V. Quine.* D. Reidel, Dordrecht. 53–68.

[Church and Gale, 1991] Church, Kenneth W. and Gale, William A. 1991. A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams. *Computer Speech and Language* 5:19–54.

[Church and Mercer, 1993] Church, K. W. and Mercer, R. L. 1993. Introduction to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics* 19(1):1–24.

[Church, 1993] Church, Kenneth W. 1993. Char_align: A Program for Aligning Parallel Texts at the Character Level. In *Proceedings of the 31rd Annual Meeting of the Association for Computational Linguistics*, Ohio State University, Columbus, Ohio.

[CL93, 1993] Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics* 19(1).

[Cover and Thomas, 1991] Cover, Thomas M. and Thomas, Joy A. 1991. *Elements of Information Theory.* John Wiley and Sons, Inc., New York.

[Cucerzan and Yarowsky, 1999] Cucerzan, Silviu and Yarowsky, David 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Proceedings of the Workshop on Very Large Corpora at the Conference on Empirical Methods in NLP.*

[Dharanipragada *et al.*, 1999] Dharanipragada, S.; Franz, M.; McCarley, J. S.; Roukos, S.; and Ward, T. 1999. Story Segmentation and Topic Detection in the Broadcast News Domain. In *Proceedings DARPA Broadcast News Workshop*, Herndon, VA. 65–68.

[Doddington, 1998] Doddington, George 1998. The Topic Detection and Tracking Phase 2 (TDT2) evaluation plan. In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA. 223–229.

[Fukumoto *et al.*, 1998] Fukumoto, J.; Shimohata, M.; Masui, F.; and Saski, M. 1998. Description of the Oki System as Used for MET-2. In *Proceedings of the MUC-7.*

[Garigliano *et al.*, 1998] Garigliano, R.; Urbanowicz, A.; and Nettleton, D. J. 1998. University of Durham: Description of the LOLITA system as Used in MUC-7. In *Proceedings of the MUC-7.*

[Good, 1953] Good, I. J. 1953. The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika* 40:237–264.

[Grishman and Sundheim, 1996] Grishman, Ralph and Sundheim, Beth 1996. "Message Understanding Conference-6: A Brief History". In *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark.

[Grishman, 1998] Grishman, Ralph 1998. "Information Extraction and Speech Recognition". In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA.

[Grosz and Sidner, 1986] Grosz, B. and Sidner, C. 1986. Attention, Intention, and the Structure of Discourse. *Computational Linguistics* 12(3):175–204.

[Hakkani *et al.*, 1998] Hakkani, Dilek Z.; Tür, Gökhan; Mitamura, Teruko; 3rd, Eric H. Nyberg; ; and Oflazer, Kemal 1998. Prototype Interlingua-Based MT System from English to Turkish. In *Proceedings of the AMTA-98.*

[Hakkani-Tür and Oflazer, 2000] Hakkani-Tür, D. and Oflazer, K. 2000. Statistical Morphological Disambiguation for Agglutinative Languages. In *Proceedings of the 18th International Conference on Computational Linguistics*, Saarbrucken, Germany.

[Hakkani-Tür *et al.*, 1999] Hakkani-Tür, Dilek; Tür, Gökhan; Stolcke, Andreas; and Shriberg, Elizabeth 1999. Combining Words and Prosody for Information Extraction from Speech. In *Proceedings of the 6th European Conference on Speech Communication and Technology*, volume 5, Budapest. 1991–1994.

[Hakkani-Tür, 2000] Hakkani-Tür, Dilek Z. 2000. *Statistical Language Modeling for Turkish.* Ph.D. Dissertation, Department of Computer Engineering, Bilkent University, Ankara, Turkey.

[Hankamer, 1989] Hankamer, Jorge 1989. Lexical Representation and Process. In Marslen-Wilson, W., editor 1989, *Morphological Parsing and the Lexicon.* The MIT Press.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. 1979. A K-Means Clustering Algorithm. *Applied Statistics* 28:100–108.

[Hearst, 1994] Hearst, M. A. 1994. Multi-paragraph Segmentation of Expository Text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, New Mexico State University, Las Cruces, NM. 9–16.

[Hearst, 1997] Hearst, Marti A. 1997. TexTiling: Segmenting text info multi-paragraph subtopic passages. *Computational Linguistics* 23(1):33–64.

[Hobbs *et al.*, 1996] Hobbs, J.; Appelt, D.; Bear, J.; Israel, D.; Kameyama, M.; Stickel, M.; and Tyson, M. 1996. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Roche, and Shabes, , editors 1996, *Finite State Devices for Natural Language Processing.* MIT Press, Cambridge, MA. 383–406.

[Humphreys *et al.*, 1998] Humphreys, K.; Gaizauskas, R.; Azzam, S.; Huyck, C.; Mitchell, B.; Cunningham, H.; and Wilks, Y. 1998. University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. In *Proceedings of the MUC-7.*

[İmla Kılavuzu, 1996] Türk Dil Kurumu.

[Jelinek *et al.*, 1975] Jelinek, Frederick; Bahl, L. R.; and Mercer, R. L. 1975. Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech. *IEEE Transactions on Information Theory* IT-21:250–256.

[Jelinek, 1968] Jelinek, Frederick 1968. *Probabilistic Information Theory.* McGraw Hill.

[Jelinek, 1998] Jelinek, Frederick 1998. *Statistical Methods for Speech Recognition.* The MIT Press.

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J. H. 2000. *Speech and Language Processing.* Prentice-Hall.

[Karttunen, 1993] Karttunen, L. 1993. Finite State Lexicon Compiler. Technical Report ISTL-NLTT-1993-04-02, XEROX Palo Alto Research Center.

[Katz, 1997] Katz, S. 1997. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing* 35(3):400–401.

[Klavans and Resnik, 1996] Klavans, Judith L. and Resnik, Philip, editors 1996. *The Balancing Act.* The MIT Press.

[Kozima, 1993] Kozima, H. 1993. Text Segmentation Based on Similarity between Words. In *Proceedings of the 31rd Annual Meeting of the Association for Computational Linguistics*, Ohio State University, Columbus, Ohio. 286–288.

[Krupka and Hausman, 1998] Krupka, G. R. and Hausman, K. 1998. IsoQuest Inc.: Description of the $NetOwl^{TM}$ Extractor System as Used for MUC-7. In *Proceedings of the MUC-7.*

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. 1951. On Information and Sufficiency. *Annals of Mathematical Statistics* 22:76–86.

[Kuruöz, 1994] Kuruöz, İlker 1994. Tagging and Morphological Disambiguation of Turkish Text. Master's thesis, Department of Computer Engineering and Information Science, Bilkent University, Ankara, Turkey.

[Lin, 1998] Lin, D. 1998. Using Collocation Statistics in Information Extraction. In *Proceedings of the MUC-7.*

[Manning and Schütze, 1999] Manning, Christopher D. and Schütze, Hinrich 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press.

[Mikheev *et al.*, 1998] Mikheev, A.; Grover, C.; and Moen, M. 1998. Description of the LTG System Used for MUC-7. In *Proceedings of the MUC-7.*

[Mikheev *et al.*, 1999] Mikheev, A.; Moens, M.; and Grover, C. 1999. Named Entity Recognition without Gazetteers. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics.*

[Miller *et al.*, 1998] Miller, Scott; Crystal, Michael; Fox, Heidi; Ramshaw, Lance; Schwartz, Richard; Stone, Rebecca; Weischedel, Ralph; and Annotation Group, the 1998. Algorithms that Learn to Extract Information; BBN: Description of the SIFT System as Used for MUC-7. In *Proceedings of the MUC-7.*

[MUC, 1995] MUC-6. *Proceedings of the MUC-6.*

[MUC, 1998] MUC-7. *Proceedings of the MUC-7.*

[NIS, 1998] NIST, National Institute of Standards and Technology. *The 1998 Hub-4 Evaluation Plan for Recognition of Broadcast News, in English.*

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

[Oflazer and Tür, 1996] Oflazer, Kemal and Tür, Gökhan 1996. Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morphological Disambiguation. In Brill, Eric and Church, Kenneth, editors 1996, *Proceedings of the ACL-SIGDAT Conference on Empirical Methods in Natural Language Processing.*

[Oflazer and Tür, 1997] Oflazer, Kemal and Tür, Gökhan 1997. Morphological Disambiguation by Voting Constraints. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid.

[Oflazer, 1993] Oflazer, Kemal 1993. Two-level Description of Turkish Morphology. *Literary and Linguistic Computing* 8(3).

[Oflazer, 1996] Oflazer, Kemal 1996. Error-Tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. *Computational Linguistics* 1(4).

[Oflazer, 1999] Oflazer, Kemal 1999. Dependency Parsing with an Extended Finite State Approach. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland, College Park, MD.

[Palmer and Finin, 1990] Palmer, Martha and Finin, Tim 1990. Workshop on the Evaluation of Natural Language Processing Systems. *Computational Linguistics* 16(3):175–181.

[Palmer and Hearst, 1997] Palmer, David D. and Hearst, Marti A. 1997. Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics* 23(2):241–267.

[Passonneau and Litman, 1997] Passonneau, Rebecca J. and Litman, Diane J. 1997. Discourse Segmentation by Human and Automated Means. *Computational Linguistics* 23(1):103–139.

[Ponte and Croft, 1997] Ponte, J. M. and Croft, W. B. 1997. Text Segmentation by Topic. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, Pisa, Italy.

[Price, 1996] Price, Patti 1996. "Combining Linguistic with Statistical Methods in Automatic Speech Understanding". In Klavans, Judith L. and Resnik, Philip, editors 1996, *The Balancing Act*. The MIT Press. 119–133.

[Quinlan, 1986] Quinlan, J. R. 1986. Induction of Decision Trees. *Machine Learning* 1(1):81–106.

[Rabiner and Juang, 1986] Rabiner, L. R. and Juang, B. H. 1986. An Introduction to Hidden Markov models. *IEEE ASSP Magazine* 3(1):4–16.

[Reynar and Ratnaparkhi, 1997] Reynar, Jeffrey C. and Ratnaparkhi, Adwait 1997. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing*, Washington,D.C.

[Reynar, 1994] Reynar, Jeffrey C. 1994. An automatic method of finding topic boundaries. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, New Mexico State University, Las Cruces, NM. 331–333.

[Reynar, 1998] Reynar, J. 1998. *Topic Segmentation: Algorithms and Applications.* Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA.

[Reynar, 1999] Reynar, Jeffrey C. 1999. Statistical Models for Topic Segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland, College Park, MD. 357–364.

[Rivlin *et al.*, 1999] Rivlin, Ze'ev; Bolles, Robert; Appelt, Douglas; Cheyer, Adam; Hakkani-Tür, Dilek Z.; Israel, David; Julia, Luc; Martin, David; Myers, Greg; Nitz, Ken; Sabata, Bikash; Sankar, Ananth; Shriberg, Elizabeth; Sonmez, Kemal; Stolcke, Andreas; and Tür, Gökhan 1999. MAESTRO: Conductor of Multimedia Analysis Technologies. *Communications of the ACM (invited submission).*

[Rosenfeld, 1994] Rosenfeld, Ronald 1994. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh. Technical Report CMU-CS-94-138.

[Shannon, 1948] Shannon, Claude E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27:379–423 and 623–656.

[Shriberg *et al.*, 2000] Shriberg, Elizabeth; Stolcke, Andreas; Hakkani-Tür, Dilek; and Tür, Gökhan 2000. Prosody-Based Automatic Segmentation of Speech into Sentences and Topics. *Speech Communication.* To appear.

[Stolcke *et al.*, 1998] Stolcke, Andreas; Shriberg, Elizabeth; Bates, Rebecca; Ostendorf, Mari; Hakkani, Dilek; Plauché, Madelaine; Tür, Gökhan; and Lu, Yu 1998. Automatic Detection of Sentence Boundaries and Disfluencies based on Recognized Words. In Robert H. Mannell and Jordi Robert-Ribes, , editor 1998, *Proceedings of the International Conference on Spoken Language Processing*, volume 5, Sydney. Australian Speech Science and Technology Association. 2247–2250.

[Stolcke *et al.*, 1999] Stolcke, Andreas; Shriberg, Elizabeth; Hakkani-Tür, Dilek; Tür, Gökhan; Rivlin, Ze'ev; and Sönmez, Kemal 1999. Combining Words and

Speech Prosody for Automatic Topic Segmentation. In *Proceedings DARPA Broadcast News Workshop*, Herndon, VA. 61–64.

[Stolcke, 1999] Stolcke, Andreas 1999. SRILM—the SRI language modeling toolkit. http://www.speech.sri.com/projects/srilm/.

[Tür *et al.*, 2000] Tür, Gökhan; Hakkani-Tür, Dilek; Stolcke, Andreas; and Shriberg, Elizabeth 2000. Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation. *Computational Linguistics*.

[Tür, 1996] Tür, Gökhan 1996. Using Multiple Sources of Information for Constraint-Based Morphological Disambiguation. Master's thesis, Department of Computer Engineering and Information Science, Bilkent University, Ankara, Turkey.

[van Mulbregt *et al.*, 1998] Mulbregt, P.van; Carp, I.; Gillick, L.; Lowe, S.; and Yamron, J. 1998. Text Segmentation and Topic Tracking on Broadcast News Via a Hidden Markov model approach. In Robert H. Mannell and Jordi Robert-Ribes, , editor 1998, *Proceedings of the International Conference on Spoken Language Processing*, volume 6, Sydney. Australian Speech Science and Technology Association. 2519–2522.

[Viterbi, 1967] Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* 13:260–269.

[Wayne, 1998] Wayne, Charles L. 1998. "Topic Detection and Tracking (TDT) Overview and Perspective". In *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA.

[Yamron *et al.*, 1998] Yamron, J.P.; Carp, I.; Gillick, L.; Lowe, S.; and Mulbregt, P.van 1998. A Hidden Markov model approach to text segmentation and event tracking. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, Seattle, WA. 333–336.

[Young and Bloothooft, 1997] Young, S. and Bloothooft, G., editors 1997. *Corpus-Based Methods In Language and Speech Processing*. Kluwer Academic Publishers.

[Yu *et al.*, 1998] Yu, Shihong; Bai, Shuanhu; and Wu, Paul 1998. Description of the Kent Ridge Digital Labs System Used for MUC-7. In *Proceedings of the MUC-7*.