SPEECH RETRIEVAL FOR TURKISH BROADCAST NEWS

by

Sıddıka Parlak

B.S., in E.E., Boğaziçi University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

I am very thankful to my advisor, Assist. Prof. Murat Saraçlar, for his great support, kind behavior and interest in this thesis. Without discussions with him and his outside help at any time, this work could not be accomplished. I will always appreciate his guidance especially at the very challenging moments.

I would like to thank Prof. Bülent Sankur for his motivation and outside help. His advicement was an invaluable guide for me during my undergraduate and graduate education. I would also like to thank Assist. Prof. Hakan Erdoğan for his gentle behavior and participation to my thesis committee.

Special thanks also to my friends at BUSIM for not only their contribution to the testing process of my project but also their great support and friendship which always gave me the power to endure the difficulties of the problem.

I would like to express my gratitude to TUBITAK project number 105E102 which funded my project and provided the financial means for this research.

I would also like to thank to my family for their endless understanding and love which gave me a great support.

ABSTRACT

SPEECH RETRIEVAL FOR TURKISH BROADCAST NEWS

Speech retrieval is a recently emerging field of information retrieval, in which the information is spoken, instead of written. So far, spoken information retrieval has been studied in several languages. In this thesis, we concentrate on the retrieval of Turkish Broadcast News. We implement two tasks: Spoken Term Detection (STD) and Spoken Document Retrieval (SDR). Although they both combine Automatic Speech Recognition (ASR) and Information Retrieval (IR) techniques to retrieve spoken data, their main goals are different. STD retrieves specific occurrences and requires an exact match, while SDR retrieves related documents and cares more about context.

Automatic transcription and retrieval of speech is more complicated in agglutinative languages because a standard size recognition vocabulary is able to cover only a limited portion of the language. A common solution is segmenting the words into subwords and using subwords units in recognition. We employed grammatical and statistical subword units in recognition and indexing for STD. Best scores are obtained via combining word and statistical subword based approaches. Word segmentation algorithms are also useful in SDR since stems bear the meaning and provide a better representation of context. Experiments showed that stemming improves SDR performance but the segmenting methods do not have a significant difference. We also studied language-independent ASR errors. Indexing the alternative ASR hypotheses, as well as the best one, was shown to be effective on the STD task. Results are presented on our Turkish Broadcast News Corpus.

ÖZET

TÜRKÇE HABER PROGRAMLARI İÇİN KONUŞMA GERİ GETİRİMİ

Son yıllarda, konuşma geri getirimi bilgi geri getiriminin bir alt dalı olarak gelişmeye başlamıştır. Alışılmışın aksine bilgi kaynağı yazılı değil, konuşma halindedir. Bu tezde, Türkçe Haber Programlarının geri getirimi üzerine çalışılmıştır. Bu amaçla iki sistem geliştirilmiştir: Konuşulan Terimlerin Saptanması (KTS) ve Konuşulan Dökümanların Geri Getirilmesi (KDGG). Her iki sistem de Otomatik Konuşma Tanıma ve Bilgi Geri Getirimi tekniklerini birleştirmektedir fakat ana hedefleri farklıdır. KTS sözcüklerin görülme zamanlarını bulmayı amaçlar ve tam olarak örtüşmeyi esas alır. KDGG ise anahtar sözcükler ile ilgili dökümanları bulmayı hedefler ve daha çok içeriğe dayalıdır.

Konuşmanın otomatik olarak yazıya çevrilmesi sondan eklemeli diller için daha karmaşıktır çünkü normal boyutlardaki bir dağarcık dilin sadece belli bir kısmını kapsayabilmektedir. Sıkça uygulanan bir çözüm kelimeleri kelime-altı birimlere ayırmak ve tanımada kelime-altı birimleri kullanmaktır. Bu çalışmada, KTS için biçimbilimsel ve istatistiksel kelime-altı birimler kullanılmıştır. En iyi sonuçlar kelime ve kelime-altı tabanlı yaklaşımların birlikte kullanılması ile elde edilmiştir. Kelime bölütleme algoritmaları KDGG için de oldukça önemlidir çünkü kelime kökleri anlam yönünden daha belirleyicidir. KDGG deneyleri kökleştirmenin başarımı iyileştirdiğini ancak yöntemler arasında önemli bir fark olmadığını göstermiştir. Ek olarak, dilden bağımsız konuşma tanıma sorunları üzerinde de durulmuştur. En iyi hipotez yerine diğer olası hipotezlerin de kullanılması KTS için başarılı sonuçlar vermiştir. Sonuçlar tarafımızca toplanan ve düzenlenen Türkçe Haber Programları Verisi üzerinde sunulmuştur.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

a_i	i'th vector of the acoustic vector sequence
a_{ij}	Transition probability from state i to state j
A	Acoustic input sequence
$A(q_k)$	Number of retrieved documents for query q_k
$b(O_i)$	Observation probability of vector O_i
$C(q_k)$	Number of correctly retrieved documents for query q_k
$C(w_i, ..., w_j)$	Count of word sequence $w_i, ..., w_j$
$C(l\|\pi)$	Count of label l in path π
d_i	Document i
$EC(l)$	Expected count of label l
idf_j	Inverse document frequency of term j
L	Language
n_j	Number of documents in which term j occurs
N	Total number of documents in the collection
O_i	i'th vector of the observed vector sequence
P_{FA}	False alarm probability
$P_G(w_i...w_j)$	Probability of the word sequence $w_i...w_j$, calculated by the general language model
P_{miss}	Miss probability
$P_S(w_i...w_j)$	Probability of the word sequence $w_i...w_j$, calculated by the specific language model
q_k	query k
Q	Total number of the queries in the system
$R(q_k)$	Number of occurrences of query q_k in reference transcripts
tf_{ij}	Frequency of term j in document i
tf_aug_{kj}	Augmented term frequency of term j in query k
$th(q_k)$	Optimum detection threshold of query q_k
T	Total number of indexing units in the system
T_{speech}	Total amount of speech

w_i	i'th word of the word sequence
w_{ij}	Weight of term j in document/query i
W	Word sequence
$\hat{W}$	Most probable word sequence
β	Cost of the false alarm probability
θ	Global detection threshold
π	A path in the lattice

ASR	Automatic Speech Recognition
ATWV	Actual Term Weighted Value
BN	Broadcast-News
BPREF	Binary Preference
DET	Detection-Error Tradeoff
FSM	Finite State Machine
G-SE	Grammatical Stem-Ending
HI	Hearing Impaired
HMM	Hidden Markov Model
IR	Information Retrieval
IRef	In-Reference
IV	In-Vocabulary
LM	Language Model
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Mean Average Precision
MAPSSWE	Matched Pairs Sentence Segment Word Error
MDL	Minimum Description Length
MFCC	Mel Frequency Cepstral Coefficients
MTWV	Maximum Term Weighted Value
OORef	Out-of-Reference
OOV	Out-of-Vocabulary
OOV-t	OOV rate of the test corpus
OOV-q	OOV rate of the query set

PLP	Perceptual Linear Predictive
ROC	Receiver Operating Characteristic
S-SE	Statistical Stem-Ending
SD	Speaker Dependent
SDR	Spoken Document Retrieval
SI	Speaker Independent
STD	Spoken Term Detection
TREC	Text Retrieval Conference
VSM	Vector Space Model
WFSA	Weighted Finite State Automata
WER	Word Error Rate

1. INTRODUCTION

Recently, the improvements in data storage and processing have brought about an increase in the amount audio-visual archives. For example, "Youtube" is a well-known video sharing website where billions of videos are uploaded and watched by internet users. Unfortunately, annotations are not usually available for this type of data because labeling is an expensive task. Extracting information by listening/watching the whole file is extremely ineffective. As a result, providing access to this huge information source is a challenge and requires sophisticated methods.

Speech Retrieval integrates Automatic Speech Recognition (ASR) and Information Retrieval (IR) techniques to access spoken data. MIT Lecture Browser is among the many applications of speech retrieval. This is a publicly available search engine to retrieve lectures from MIT Lecture Corpus, which contains more than 200 lectures. Users can search lectures for any term they want and then play the relevant sections using a web interface.

AT&T's VoIP Meeting Service is a similar tool which is used for setting up and running meetings. With this tool, it is possible to record a conference and find the important parts, instead of listening to the whole meeting.

Turkish Sign Language Tutoring Tool is a relatively novel application of speech retrieval. The task is to display the sign corresponding to a user query. Turkish Broadcast News for the Hearing Impaired are perfectly appropriate as the source since the speaker makes the signs as she speaks. The signs are located with the help of speech and the related portion of the video is displayed to the user [49].

Traditional speech retrieval systems first convert speech to text using automatic speech recognizers, then apply text IR algorithms on the ASR output. However, even state of the art ASR systems can not generate perfect transcriptions. Thus, text IR methods are inadequate to index ASR output, especially when the ASR system has low

accuracy. Possible reasons of ASR inaccuracy include training mismatch, poor acoustic conditions and Out-of-Vocabulary (OOV) words. Indexing the alternative hypotheses as well as the best one makes the system more robust to ASR errors. However, the words that are not in the recognition vocabulary still can not be recognized. This problem is more severe for Turkish, because of agglutinativity. Using subword units is a common approach to recognize OOV words. Subword units can be directly used in indexing, in addition to language modeling in ASR.

In this work, we will be concentrating on two speech retrieval tasks: Spoken Term Detection and Spoken Document Retrieval. STD aims to locate the occurrences of a query. It is different from traditional keyword spotting, since the queries are not known prior to search. The Turkish Sign Language Tutoring Tool is an application of STD. Another application might be monitoring, where the aim is to detect the occurrences of particular terms. SDR is our second task, which retrieves the documents (news stories) related to a query. Unlike STD, SDR is not totally based on the term occurrence. The topic of the document and its relation to the query have the major importance. From this point of view, SDR is more analogous to the traditional text based IR. Applications of SDR include retrieving voicemail messages, news stories or academic lectures related to the user request.

We construct STD and SDR systems for retrieval of Turkish Broadcast News. They employ the same speech recognizer in the first stage and differentiate in indexing and retrieval phases. In order to alleviate the OOV problem of STD, grammatical and statistical word segmenting algorithms are applied. The same approaches are used in SDR as well, for stemming purposes. Since ASR output is erroneous, we make use of expanded hypotheses, such as lattices and confusion networks. This work is the first speech retrieval study in Turkish. The experimental results are presented on our Turkish Broadcast News Corpus.

1.1. Thesis Outline

This thesis is organized as follows: In Chapter 2, first, we give an overview of the major components of speech retrieval: ASR and IR. Next, their integration is described and STD and SDR tasks are defined. Chapter 3 consists of a brief information about Turkish Broadcast News corpus. We explain our STD system in Chapter 4 and present the experimental results. SDR system description and experiments are introduced in Chapter 5. Finally, we conclude our work and results in Chapter 6.

Appendix A includes an overview of Finite State Machines. Significance testing results of STD experiments are given in Appendix B and Appendix C. SDR topics are presented in Appendix D.

2. BACKGROUND

As mentioned previously, speech retrieval systems integrate two basic components: ASR and IR. However, applying text based IR methods on the speech recognizer transcriptions is not so straightforward. In this section, first we will give an overview of ASR and IR components. Next, we will explain how these systems are brought together and the challenges introduced by their combination.

2.1. Automatic Speech Recognition

The aim of automatic speech recognition is to build systems for mapping an acoustic speech signal to a sequence of words. Although automatic transcription of speech is an unsolved problem, current systems are able to handle many tasks. A major application area is human-computer interaction. Speech is a more natural interface than keyboard and it may be the only way of interaction for hands-busy or eyes-busy applications. Some operations over the telephone, such as voice dialing and call routing, incorporate automatic speech recognizers. In law and medicine, dictation systems are commonly used. More recent technologies, spoken audio indexing and search systems make use of ASR systems.

Speech recognition problem can be discussed in various dimensions such as vocabulary size, continuity and speaker dependency. Large vocabulary continuous speech recognition (LVCSR) is a major field, where *large vocabulary* means a vocabulary of about 5000 to 60000 words and *continuous* means words are spoken continuously (unlike isolated word recognition, where each word is followed by a pause). Speaker independent (SI) systems are able to recognize speech from people whose speech the system has never processed before whereas speaker dependent (SD) systems can recognize only speech from people whose speech the system has processed before [1].

Primary goal of statistical speech recognition is to find the most probable word sequence $\hat{W} = w_1 w_2 ... w_n$ in language L, given some acoustic input, $A = a_1 a_2 ... a_n$. This

intuition can be expressed as follows:

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} P(W|A) \tag{2.1}$$

Applying the Bayes rule we have:

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \frac{P(A|W)P(W)}{P(A)} \tag{2.2}$$

Since the maximum is found among all possible word sequences and the acoustic sequence probability is same for all of them, we can ignore the *P(A)* term.

$$\hat{W} = \underset{W \in L}{\operatorname{argmax}} \frac{P(A|W)P(W)}{P(A)} = \underset{W \in L}{\operatorname{argmax}} P(A|W)P(W) \tag{2.3}$$

In equation 2.3, P(W) is the prior probability, computed by the language model, $P(A|W)$ is the observation likelihood, computed by the acoustic model and maximum is found via Viterbi decoding [1].

$$\hat{W} = \underbrace{\underset{W \in L}{\operatorname{argmax}}}_{\text{decoding}} \underbrace{P(A|W)}_{\text{acoustic model}} \underbrace{P(W)}_{\text{language model}} \tag{2.4}$$

2.1.1. Acoustic Model

2.1.1.1. Feature Extraction. In speech recognition, acoustic data is represented with a sequence of vectors. The process of obtaining these vectors is called *feature extraction.*

As the first step of feature extraction, the input sound is digitized (sampled and quantized). Common sampling rates are 8kHz for telephone speech and 16kHz for direct microphone output. Once the waveform has been digitized, it is converted into a set of feature vectors. The most popular set of features are cepstrum coefficients,

such as Mel-Frequency Cepstrum Coefficients (MFCC) or Perceptual Linear Predictive (PLP) Coefficients, which are consistent with human hearing. The mel scale is linear in low frequency range (< 1000Hz), and logarithmic above 1000Hz, which approximates the human auditory system. The cepstral features are obtained by taking the inverse transform of the log of the filterbank parameters and applying discrete cosine transform [2],[1]. Details of the MFCC feature extraction can be found in [3].

In order to capture the dynamic nature of the speech signal, first and second derivatives of the features are also added to the vector. The resulting feature vector consists of 12 cepstrum coefficients plus the log energy and first and second order derivatives, i.e. a total of 39 components [2].

2.1.1.2. Hidden Markov Models. The probabilities on feature vectors are computed by the Hidden Markov Model (HMM). In an observable Markov model, each state corresponds to a deterministic event. Whereas in an HMM, observations are probabilistic functions of the state.

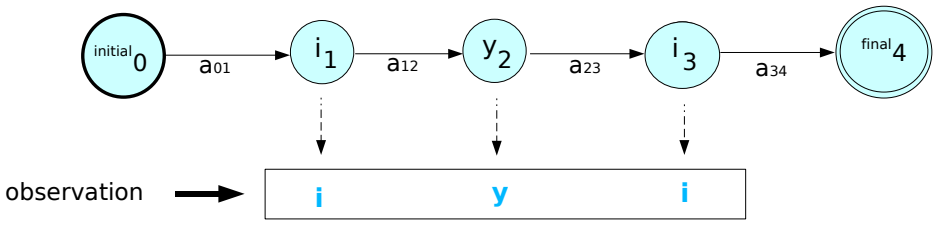

Figure 2.1. An observable markov model

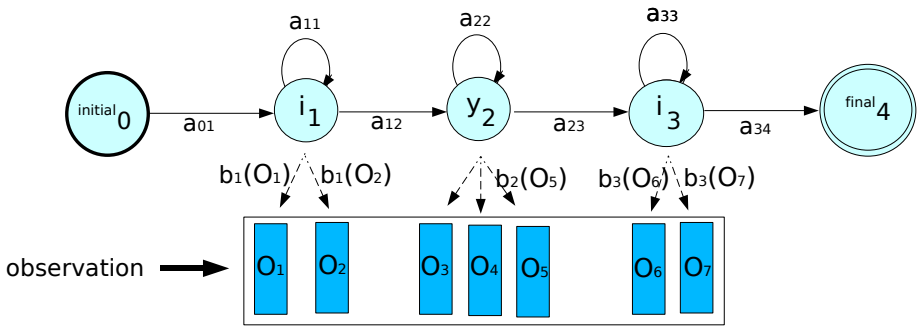

Figure 2.2. A hidden markov model

Figures 2.1 and 2.2 illustrate the difference. In Figure 2.1, a simple pronunciation model is depicted. Note that the observed phone sequence is directly determined by the phone state sequence. However in Figure 2.2, observed sequence is not a phone sequence but a vector sequence, obtained from the acoustic signal via feature extraction. Additionally, various output vectors can be emitted from the same state. For example, at state "1", the probability of observing "O_1" is b(O_1) and the probability of observing "O_2" is $b(O_2)$. (These probabilities are "1" or "0" for the observable Markov case.)

Therefore, an HMM is defined by observation likelihoods in addition to the observable Markov model parameters: states and transition probabilities. The three basic problems associated with an HMM are:

1. Given the parameters of the model, computing the probability of a particular output sequence, and the probabilities of the hidden state values given that output sequence. This problem is solved by the forward-backward algorithm.
2. Given the parameters of the model, finding the most likely sequence of hidden states that could have generated a given output sequence. This problem is solved by the Viterbi algorithm. (The Viterbi decoding mentioned in Section 2.1)
3. Given an output sequence or a set of such sequences, finding the most likely set of state transition and output probabilities. In other words, discovering the parameters of the HMM given a dataset of sequences. This problem is solved by the Baum-Welch algorithm. (The acoustic model training process)

In state of the art ASR systems, this basic HMM model is improved by taking the contexts of the phones into account. Since vocal articulators can not change shape very fast, the articulation of a phone is affected by the neighboring phones. Based on this idea, context dependent models are used to improve the modelling accuracy and recognition performance. A context dependent triphone model is trained with a sequence of three phones; each phone is modeled in the context of the previous and the next phone. However, use of triphones causes a sharp increase in the number of models ($29^3 = 24389$ models for Turkish) and some triphones might be missing in the training data. As the solution, decision trees are commonly used to find and cluster

similar triphones. It is also possible to create a tree for each HMM state (instead of a phoneme) position of each phoneme. State-based clustering has been shown to provide further improvements.

2.1.2. Language Model

A statistical language model estimates the probability of a word sequence in the language. In speech recognition, the prior probability in equation 2.4 (i.e. *P(W)*) is computed by the N-gram language model. N-grams are statistical models, which predict the next word from the previous N-1 words.

The probability of a word sequence is expressed as follows:

$$P(W) = P(w_1 w_2 ... w_n) \tag{2.5}$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1 w_2)...P(w_n|w_1 w_2 ... w_{n-1}) \tag{2.6}$$

$$= \prod_{k=1}^{n} P(w_k|w_1 w_2 ... w_{k-1}) \tag{2.7}$$

In equation 2.6, $P(w_k|w_1 w_2 ... w_{k-1})$ is the probability of w_k is spoken after the word sequence $w_1 w_2 ... w_{k-1}$. This sequence is also named as *history*. Although considering the whole history is more accurate, a longer string tends to appear rare in the corpus. This reduces the reliability of the estimate. The solution is fixing the history length to a number of words and use the following approximation:

$$P(w_k|w_1 w_2 ... w_{k-1}) \approx P(w_k|w_{k-N+1} ... w_{k-1}) \tag{2.8}$$

Thus, the probability of w_k given the whole history can be approximated by the probability given only the previous $N-1$ words.

If $N = 1$ (i.e. all words are independent), language model is a unigram. If $N = 2$ (i.e. the word is dependent on only its previous word.), LM is a bigram. If $N = 3$ (i.e. the word is dependent on only its previous two words.), LM is a trigram. Higher order N-grams are better in predicting the next word, however they require a larger training corpus and larger disk space.

The probability of a word given the word history is computed via estimating the relative frequencies in the corpus. For example, for the trigram approach:

$$P(w_k|w_{k-2}w_{k-1}) = \frac{C(w_{k-2}w_{k-1}w_k)}{C(w_{k-2}w_{k-1})} \tag{2.9}$$

where $C(w_{k-2}w_{k-1})$ is the count of word sequence $w_{k-2}w_{k-1}$ [1].

The main problem of N-gram language modeling is the data sparseness. If the training corpus is not sufficiently large, very small or zero probabilities may be assigned to possible word sequences. Various smoothing methods are used to cope with data sparseness such as Add-One, Witten-Bell, Katz and Kneser-Ney smoothing.

Domain of the corpus is another important factor in building a reliable model. For example, the word sequence *"günaydın sayın seyirciler"* is frequent in broadcast news however it does not appear on the internet, which is a huge text source. Utilizing both the size of web corpus and domain of broadcast news corpus is possible via language model adaptation methods.

2.1.2.1. Subword Based Language Modeling. Since an ASR system is able to recognize only the words that are in the recognition vocabulary, coverage is a critical point in language modeling. Agglutinative languages like Turkish are disadvantageous about this issue due to the huge vocabulary explosion. In Turkish, it is possible to obtain many different words from a single root, by adding suffixes consecutively. An example word is shown below:

```
EV + im + de + ki + ler + den : evimdekilerden
```

This word consists of one root and 5 suffixes. Its meaning in English is "(one) of those in my house" which includes 5 words.

As a result, word-based models result in high OOV rate in agglutinative languages; OOV rate of an 50k vocabulary is about 10% for Turkish while it is about 1% for English. Increasing the vocabulary size is a solution but it requires more memory and processing power. In addition, high vocabulary size causes the language model to be non-robust due to the data sparseness. Much more amount of training data is needed to model a large number of words. Therefore, the feasible solution is using subwords as the language modeling unit, instead of words. Some commonly used subword units are listed below:

1. Linguistically Derived Units
    - Phones: Phones are the smallest identifiable units found in a stream of speech.
    - Syllables: Syllables are units of sound each of which is composed of a vowel (as the center) and consonants around the vowel.
    - Morphemes: Morphemes are grammatical units (stem, prefix and suffix) of a word that are extracted by a morphological parser. Since prefixes are extremely rare in Turkish (usually occur in borrowed words), first morpheme is the stem and remaining morphemes are the suffixes.
    - Grammatical Stem-Endings (G-SE): As long as the OOV rate is low, longer recognition units work better than smaller length units. The reason might be the ability of word-based models to capture the relation between a longer sequence of words. Namely, a trigram word based model covers three words, however a trigram morpheme based model covers three morphemes, which is probably shorter than three words. Increasing the order of the N-gram is possible but it requires more processing power and memory. On the other hand, the inflective structure of Turkish necessitate subword based models.

As a solution, the suffixes can be grouped into a single unit, called ending. Since a Turkish word usually does not include a prefix, this approach results in maximum 2 subword units for each word. Therefore grammatical stem ending units are favorable for Turkish [4]. An example parse is shown below.

$$\underbrace{ev}_{stem} + \underbrace{im + de + ki + ler + den}_{ending}$$

2. Statistically Derived Units
   - Morphs: Morphs are extracted using an algorithm that is based on the Minimum Description Length (MDL) principle. The algorithm discovers subword units in an unsupervised manner from a training corpus. Next, the whole corpus is segmented into morphs via Viterbi algorithm using the generated morph types [4]. In order to eliminate huge number of morph types, rare words are not used for morph generation [5].
   - Statistical Stem-Endings (S-SE): Statistical stem-endings are derived based on the idea of G-SEs: If the OOV rate is low, longer recognition units demonstrate better performance. The building blocks of S-SEs are morphs, instead of morphemes. Namely, a word is segmented into morphs, then first morph is assumed to be the "stem" and remaining morphs are grouped to be the "ending" [5].
   - Particles: Particles are syllable-like data-driven units. They are determined to maximize leaving-one-out likelihood of a particle based bigram language model [6].

2.1.3. Automatic Speech Recognition Evaluation

The standard evaluation metric for speech recognition is called *word error rate* (WER). WER is the difference of hypothesized word string and the reference transcription in minimum edit distance of words. Namely, the minimum number of word substitutions, insertions and deletions necessary to convert hypothesis to the reference transcription is identified. The WER is computed using dynamic programming with

the following formula:

$$\text{WER} = 100 \cdot \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Number of words in correct transcript}} \tag{2.10}$$

2.1.4. Automatic Speech Recognition in Turkish

In Turkish, LVCSR research primarily focuses on techniques to deal with vocabulary growth.

In [7], first Turkish SI LVCSR results are presented on 1 hour of read speech. Morphology based language modeling and vocabulary adaptation methods are exploited to overcome the difficulties posed by agglutinativity. Vocabulary adaptation is shown to be more effective.

Another study aims to build a speech database for Turkish. 20 speakers are asked to read isolated words and continuous utterances. Morphology-based language models are used to reduce OOV rate. Recognition of isolated words gives promising results [8].

Morpheme-based, stem-ending-based (from morphemes) and syllable-based language modeling performances are compared with respect to coverage, OOV words, perplexity and sensitivity to context in [9]. It is concluded that, although the syllable-based model results in very low OOV rate and perplexity, morpheme and stem-ending based language models are the most appropriate ones for ASR because they have a better representation of meaning.

[10] presents the first data driven subword based language modeling for Turkish and compares with morphological units. Experiments show that, the data-driven approach outperforms the morphology-based language model. This may arise from the relatively longer recognition units of the data-driven model. Another study evaluates data driven subword based language modeling on three agglutinative languages including Turkish [11].

Rescoring with root and class based methods are applied to Turkish LVCSR in [12]. Vocabulary extension methods are also utilized to handle the OOV problem and results in an absolute decrease of 1% WER.

Since 2006, a Turkish Broadcast News corpus has been collected at Boğaziçi University. Details about this database will be explained in Chapter 3. [4] introduces morphology based surface and lexical units for Turkish ASR. According to the experimental results on the mentioned corpus, lexical stem-endings are superior than other units. Another study on the same corpus compares various language modeling units [5]. This paper also investigates the effect of vocabulary size and amount of training data in ASR.

2.2. Information Retrieval

Information Retrieval (IR) research aims to facilitate access to materials that are related to a user information request. Traditional IR research and applications are mainly based on text. Multimedia IR is a relatively new area of research. This section gives an overview of text retrieval.

An index is the main component of an IR system. Consider the index of a book. It is possible to find out the pages on a subject using the index, without going over all of the pages. Likewise, the index of an IR system keeps all of the necessary information to find related documents. The retrieval module directly uses the index, instead of scanning the whole collection. This facilitates finding the relevant documents and makes the search speed less dependent on collection size.

Construction of an index depends primarily on the retrieval type. The search engine may return documents that satisfy the information need precisely (e.g. Boolean Model) or partially (e.g. Vector Space Model-VSM). Partial matching systems rank the documents according to an estimated relevance score [13], [1]. Statistical language modeling is also applied to IR and it is an example of partial matching. This approach builds a language model for each document in the collection and computes the proba-

bility of producing the user query for each document. The documents are ranked with descending probability [14].

Additional techniques are applied on the baseline indexing+retrieval system to increase IR efficiency. Some of them are summarized below:

- Stemming: Since the words with the same stem usually indicate the same topic, it is beneficial to prune the inflected words and use directly their stems for indexing. Consider the words *deprem* (earthquake), *depremi* (... earthquake), *depreminin* (the earthquake of ...) and *depreminde* (in the earthquake of ...). Even if the query is *deprem*, a document that contains one of the above words would be useful. Some of the several stemming techniques are:
  - Indexing the first n characters of each term in the collection
  - Using a morphological parser to obtain the stem of the word (Stem of G-SE in Section 2.1.2.1)
  - Using statistical methods to segment words (Stem of S-SE in Section 2.1.2.1)

  Although the first approach seems too simple, it is shown to be very useful in [15].
- Query and Document Expansion: These methods aim to reduce the query-document mismatch. Reformulating the query using words or phrases with a similar or related meaning is called *query expansion*. Examples include searching for synonyms, morphological variants etc. In *document expansion*, the documents are augmented with related terms such as the alternative recognition hypotheses (for spoken documents).
- Dimensionality Reduction: Dimensionality reduction in the term document matrix also reduces the query-document mismatch by grouping related terms. A popular technique is *latent semantic indexing*, which employs singular value decomposition. It is also possible to use other techniques such as nonnegative matrix factorization [16].
- Stop Lists: Some words in the language are not useful for finding a relevant document, such as "için"(for) and "bu"(this) in Turkish. Because of this reason, the words in the stop lists are excluded from indexing. Stop lists reduce the size

of the index [17].

The Text Retrieval Conference (TREC) is an important initiative in IR research. For each TREC, NIST provides a test set of documents and questions. Participants run their own retrieval systems on the data, and return a list of the retrieved documents to NIST. NIST evaluates and declares the results [18]. Although the main collection is in English; Arabic, Spanish and Chinese collections are also available.

The Cross-Language Evaluation Forum (CLEF) encourages research to develop retrieval systems capable of searching over collections in various languages. Multilingual IR is important in disseminating information over several languages. It is a multidisciplinary area integrating fields such as information retrieval, natural language processing, machine translation and summarization. Over the years CLEF has expanded the coverage to multimedia in addition to text.

2.2.1. IR on Turkish Documents

IR research is not common for languages other than English, which is also the case for Turkish. Some of the work that has been done on Turkish is summarized below:

Effectiveness of four Turkish search engines are compared in [19] using 17 queries. They evaluate the coverage, novelty and recency of the engines as well as the conventional precision and recall rates. A similar study evaluates popular search engines' performances on finding Turkish documents [20]. Interestingly, local search engines are shown to perform worse than international ones on Turkish documents.

In [21], a two level morphology based stemming approach is evaluated on 6289 Turkish documents consisting of economic and political news stories. 50 queries are submitted to the OKAPI text retrieval system. Stemming is shown to be very useful in overall. However unstemmed search gives better results for part of the queries. Their explanation is the lack of disambiguation on morphological parser's output.

Several natural language processing approaches are utilized in [22] in addition to traditional morphological approaches. Each word is expanded with related words (lexico-semantical level) and part-of-speech tags (syntactic level) are used. Experiments on 615 documents and 5 queries show that syntactic information contributes to IR effectiveness whereas lexico-semantical information does not.

A recent study has provided a TREC like collection for Turkish [15]. The collection consists of 408305 documents from the newspaper Milliyet. 72 query topics are determined and relevancy of the documents are judged by human assessors. Three stemming approaches are investigated: simple word truncation (fixed prefix), successor variety method (data-driven) and lemmatizer-based stemming, which incorporates a morphological parser. Additional experiments investigate the effect of several query-document matching functions, query & document length, stop lists and collection size. Overall conclusion is that use of stopword lists and language specific stemming algorithms do not contribute much to system performance. On the other hand, longer queries and longer documents are better than short ones for retrieval.

2.2.2. Evaluation of IR Systems

2.2.2.1. Evaluation Metrics. Precision and recall are the most popular IR evaluation metrics and defined as follows: Given Q queries, let $R(q_k)$ be the number of documents in the collection that are related to the query q_k, $A(q_k)$ be the total number of retrieved documents and $C(q_k)$ be the number of correctly retrieved documents. Then:

$$\text{Precision} = \frac{1}{Q}\sum_{k=1}^{Q}\frac{C(q_k)}{A(q_k)} \qquad \text{Recall} = \frac{1}{Q}\sum_{k=1}^{Q}\frac{C(q_k)}{R(q_k)} \tag{2.11}$$

Note the trade-off between precision and recall. Recall is usually low at the high precision region (high ranked documents) since many related documents might be missed. Likewise, precision is low at the high recall region since many irrelevant documents are included. Precision-recall curves are commonly used to analyze the

relation between precision and recall rates of an IR system. Figure 2.3 shows a typical precision-recall curve.

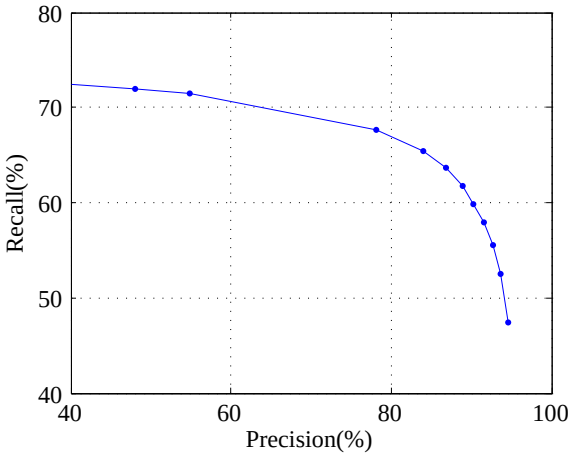

Figure 2.3. A typical precision-recall graph

Although precision and recall rates are advantageous for tracing the system behavior, sometimes it is useful to evaluate the system with a single score. Precision at the top n documents (usually 5, 10 or 20) is a commonly used metric, especially for web search engines. Mean Average Precision (MAP) is a composite measure which depends on both precision and recall. To compute MAP, precision is computed at each point in the list where a correctly retrieved document is found and all precision values are averaged. Formally:

$$\text{MAP} = \frac{1}{Q}\sum_{k=1}^{Q}\frac{1}{R(q_k)}\sum_{r=1}^{R(q_k)}\{\text{Precision when r}'\text{th relevant document is retrieved}\} \tag{2.12}$$

This corresponds to the area under the precision-recall curve.

F-measure is another metric which is the weighted harmonic mean of precision and recall. The general formula for F_β is:

$$F_\beta = \frac{(1+\beta^2)\cdot \text{Precision}\cdot \text{Recall}}{\beta^2\cdot \text{Precision} + \text{Recall}} \tag{2.13}$$

where β is the weight of recall; using $\beta < 1$ emphasizes precision whereas using $\beta > 1$ emphasizes recall. F_1 measure is the most common type, where precision and recall are equally weighted. It is also named as F-measure.

$$F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.14}$$

A novel metric to evaluate STD systems is the "Actual Term Weighted Value" which is defined in NIST STD 2006 Evaluation Plan [24] as:

$$\text{ATWV}(\theta) = 1 - \frac{1}{Q} \sum_{k=1}^{Q} \{P_{miss}(q_k, \theta) + \beta . P_{FA}(q_k, \theta)\} \tag{2.15}$$

where β is a user defined parameter to adjust the trade-off between miss and false alarm probabilities. θ is the detection threshold. ATWV is calculated for all possible θ values and the maximum is called the Maximum Term Weighted Value (MTWV). P_{miss} and P_{FA} are calculated as:

$$\text{P}_{\text{miss}}(q_k, \theta) = 1 - \frac{C(q_k, \theta)}{R(q_k)} \qquad \text{P}_{\text{FA}}(\text{q}_\text{k}, \theta) = \frac{A(q_k, \theta) - C(q_k, \theta)}{T_{speech} - C(q_k, \theta)} \tag{2.16}$$

where T_{speech} is the total amount of speech.

2.2.2.2. Relevance Assessments. To evaluate an IR system, the documents in the collection should be labeled manually as relevant or irrelevant to each of the queries in the query set. The relevancy of a document to a query is preferably determined by multiple independent judges because of the subjectivity of relevance decision. Namely, a number of assessors label document d_i as relevant or irrelevant to query q_k (binary relevance judgement). If majority of the labels are "relevant", then document d_i is assigned as relevant to query q_k and vice versa.

As the collection expands, manual labeling of the whole data gets extremely harder. For example the TREC-8 SDR track contains 21754 stories and 50 topics, which corresponds to 21754*50=1087700 binary relevance judgements.

Pooling is a way to ease the labeling task. In this approach, only the documents in the pool are labeled as relevant or irrelevant. Remaining ones are simply assumed to be irrelevant. To construct the query pools, several runs are made with various methods. For each query, top n documents of each run is added to the pool. This may introduce a bias to the evaluation since the methods used in pooling may be favored. In [15], *Binary Preference* (BPREF) metric is employed to eliminate this deficiency. BPREF is defined in [23] as follows: For a topic with $R(q_k)$ relevant documents where r is a relevant document and i is a member of the first $R(q_k)$ judged irrelevant documents retrieved by the system:

$$\text{BPREF}(q_k) = \frac{1}{R(q_k)} \sum_{r} (1 - \frac{|\text{i ranked higher than r}|}{R(q_k)}) \tag{2.17}$$

2.2.3. Vector Space Model

Vector space model (VSM) computes the similarity between a document and a query using term weights (features). Queries and documents are represented as T-dimensional vectors, $d_i(w_{i1}, w_{i2}, ..., w_{iT})$ and $q_k(w_{k1}, w_{k2}, ..., w_{kT})$, where T is the total number of indexing units in the system. Thus, the document collection can be regarded as a *term-by-document matrix*. The similarity is quantified with the cosine angle between two vectors (via Equation 2.18) and the retrieved documents are ranked using this relevance score. VSM treats the documents as bag-of-words. In a document, word or sentence ordering is not important.

$$\text{similarity}(\text{d}_\text{i}, \text{q}_\text{k}) = \frac{d_i \cdot q_k}{|d_i| \cdot |q_k|} \tag{2.18}$$

The term weight calculation is based on two components: term frequency and

inverse document frequency. Term frequency (tf) is the frequency of a term within a document and shows how well the term describes the document contents. Inverse document frequency (idf) is used to favor the terms that occur in fewer documents because terms which appear in many documents are not useful in distinguishing a relevant document from a non-relevant one. Inverse document frequency of a term is calculated as:

$$\mathrm{idf_j} = \log(\frac{N}{n_j}) \tag{2.19}$$

where N is the total number of documents in the collection and n_j is the number of documents in which term j occurs. Due to the large number of documents in many collections, this measure is usually squashed with a log function. The well known tf.idf measure is the product of tf and idf components.

$$w_{ij} = \mathrm{tf}_{ij} \times \mathrm{idf}_j \tag{2.20}$$

The tf and idf calculation may have minor variations in the literature. However, all of the techniques are based on the principal explained above. Also note that, the weights are calculated for the query and the document in the same way. However, queries are not like documents; they include very limited number of words compared to documents. This damages the reliability and usefulness of the tf component. To compensate that, the *augmented term frequency* is proposed for queries.

$$\mathrm{tf\_aug}_{ik} = (0.5 + \frac{0.5 \times tf_{ik}}{\max_i tf_{ik}}) \tag{2.21}$$

where $max_i tf_{ik}$ denotes frequency of the most frequent term in document i [1], [13].

2.3. Integration of ASR and IR: Spoken Information Retrieval

Both, ASR and IR has been extensively discussed for several decades. Their integration, on the other hand, is an emerging field in the last decades. Spoken audio has a rich content. It includes additional pieces of information which can not be represented by text, such as emotional state. However, spoken documents are not as structured as textual documents. First they need to be transcribed by automatic speech recognizers. These automatic transcripts include errors and noise (e.g. hesitations, repetitions). Even if they are mainly correct, they have a linear structure: just a long sequence of words. They are not segmented into paragraphs, do not have titles etc. This makes them more difficult to browse. Some approaches to organize spoken content, are explained below [25], [26].

1. **Named Entity Tagging:** Named entities are usually the key words in the spoken content, such as proper nouns, temporal expressions, numerical quantities etc. They may have a key role to understand the subject of the spoken document.
2. **Topic Segmentation:** The automatic transcripts are segmented into paragraphs such that each paragraph is on a particular topic.
3. **Information Extraction:** Extraction of the key information, usually by using Named Entities
4. **Summarization:** Automatically generating a summary for each paragraph (which are segmented by topic).
5. **Title Generation:** Automatically generating a title for each paragraph.
6. **Topic Analysis and Organization:** Analyzing the topic of each paragraph and organizing to facilitate browsing.

Following the organization of spoken documents, they can be retrieved via classical text retrieval methods explained in the previous section.

Several research groups developed speech retrieval systems for indexing and retrieval of voicemail messages, broadcast news, meeting and lecture recordings etc. An example is the "Rough'n'Ready" system which has been built by BBN technologies

for the retrieval of broadcast news. This system does not utilize cues from video, it is merely based on speech. Rough'n'Ready incorporates several technologies such as LVCSR, speaker segmentation, clustering and identification, name spotting, topic segmentation and classification [27]. "Speechbot" indexes audio and video files (popular talk radio, technical and financial news shows, conference video recordings) from the web [28]. Although this type of audio suffers from high WER due to the acoustic condition variability, the overall retrieval system performs well. "Speechfind" retrieves spoken documents from a historical archive (National Gallery of the Spoken Word - NGSW) spanning the 20th century. Query and document expansion methods are shown to be successful, with an improvement of 8% in absolute precision. SpeechFind search system is currently available on the web [29].

2.3.1. Spoken Term Detection

STD is a subfield of speech retrieval which locates occurrences of a query in a spoken archive. STD is an ad-hoc type retrieval; the user submits a query to the system and the search engine lists the results. The retrieved information includes the occurrence time in addition to the document name and decision (or relevance) score. From this point of view, STD has important differences from traditional text retrieval. For example, topic of the document is not important. Moreover, stemming, clustering or stopword elimination methods of IR can not be used because the query should be matched exactly. On the other hand, it is possible to evaluate STD systems with IR evaluation metrics.

The NIST STD 2006 Evaluation has initiated the STD track in three languages: Arabic, English and Mandarin. The evaluation was not only based on correct detection but also indexing and search time, index size and memory consumption. Several research groups participated in the evaluation. The SRI/OGI system achieved one of the best scores using a word+graphone system [30]. BBN system achieved the maximum accuracy in continuous telephone speech domain. Their index was based on word lattices and approximate phonetic transcripts (for OOV words) [31].

Developers of Speechbot [28], which was a publicly available system, report that the percentage of OOV queries submitted to a real engine is about 13% [32]. To emphasize the effect of OOV queries, OOV percentage is artificially increased to 50% in [32]. Word-based, phoneme-based and particle-based systems are compared. Hybrids of word and subword systems are shown to demonstrate the best performance. Further, the group evaluates the effect of query expansion on the hybrid system [33]. An OOV query is expanded to IV queries based on acoustic confusability and language model scores. This method provides an improvement of 1% in average precision.

Several other studies show the superiority of hybrid systems. In [34], phone and word cascades are shown to be useful. [35] directly combines the word and phone indexes using time stamps and allows hybrid queries to be searched in both indexes.

Using alternative hypotheses of the ASR output, in addition to the best one, is another useful approach [34], [36]. These alternative hypotheses can be in the form of lattices or confusion networks, which will be explained in Chapter 4.2.1.1. It is reported in [37] that, confusion networks perform better than lattices, in addition to their advantage of having smaller index sizes.

In [38], a two stage method is used to decrease search time. First, a set of relevant documents are retrieved via VSM. Next, the query is located only in this set of documents. They vary the size of the set with a pruning threshold in VSM stage and analyze the degrade in detection performance with increasing speed.

2.3.2. Spoken Document Retrieval

Spoken document retrieval (SDR) is the content based retrieval of spoken audio, which brings the speech recognition and information retrieval research together to access spoken information. The main purpose and challenges of the SDR task differ from STD. For SDR, content of the document is more important than exact term matching. As a result, SDR is less affected by OOV words, unlike STD. In SDR, it may be possible to find a relevant document, even if the query is OOV, using the

other query words or clustering. However this is not possible in STD, unless additional methods are used.

The TREC Spoken Document Retrieval Track was an important initiative in SDR research. The first evaluation, TREC-6 SDR (known-item retrieval), resulted in an encouraging success but the conclusions were not so reliable because of the limited amount of data. Ad-hoc retrieval of spoken documents started with TREC-7. The audio collection was extended to 557 hours for TREC-8. Retrieval was shown to be robust to recognition errors because the slope of linearity was very small. The probable reason was explained as follows: The redundancy of keywords in the spoken documents permits the relevant documents to be retrieved, even if a number of words are misrecognized [39]. Finally, TREC-9 gave extremely encouraging results even when the story boundaries are unknown. After the four evaluations (1997-known item retrieval, 1998, 1999 and 2000 ad-hoc retrieval), it was concluded that the accuracy of retrieval from speech recognizer transcripts could be very close to human reference transcripts.

The conclusion of TREC SDR Tracks, stating that the problem has been solved, was drawn mainly for English. SDR was still a challenge for agglutinative languages. The problem has been investigated for Finnish in the following papers. In [40], grammatical baseforms and morphs are indexed, instead of words. Both methods are shown to be efficient in approximating the human transcripts. Query and document expansion techniques are added to the system in their following study [41]. Finally, in [42], alternative ASR hypotheses are indexed, using morph confusion networks. Frequency of a term in a document is computed by summing the posterior probabilities or by summing the inverse ranks (inverse of the word's posterior probability rank in that word's bin). The results show that the latter term frequency calculation provides higher improvement in MAP score.

Some of the previously mentioned speech retrieval systems, such as Rough'n'Ready [27], SpeechFind [29], and CastSearch [16], are the other examples of SDR. SpeechFind and CastSearch makes use of the standard VSM model. On the other hand, Rough'n'Ready is an HMM-based IR system.

3. TURKISH BROADCAST NEWS DATABASE

A large Turkish Broadcast News database has been collected at Boğaziçi University since March 2006. News videos are recorded daily from four television channels (CNN-Türk, NTV, TRT1, TRT2) and one radio channel (VoA). TRT2 recordings also include the Broadcast News for the Hearing Impaired (HI).

The broadcast news audio is recorded automatically using a PC with a TV/Radio tuner card. A planner is included in the software of the card so that no manual control is needed to start/stop recording. The recordings that do not include news content and lack audio quality are discarded. Remaining programmes are segmented, transcribed, verified and added to the news database.

3.1. Segmentation

Most of the news programmes are segmented into utterances manually. Speech segments are further annotated with speaker and background information. Non-speech segments are excluded from this annotation and transcription. Classical Hub4 classes are used to label the background: f0 (clean speech), f1 (spontaneous speech), f2 (telephone speech), f3 (background music), f4 (degraded acoustic conditions) and fx (other). It takes about 1.5-2.5 hours to segment and annotate a 1-hour video.

Part of the HI news are segmented automatically, based on the energy. Automatic segmentations do not have labels.

3.2. Transcription

Segmented audio files are transcribed manually. The transcription guidelines are adapted from Hub4 Broadcast News transcription guidelines. Part of the transcribed data is manually checked and corrected. It takes about 4-6 hours to transcribe a 1-

hour video. The open source Transcriber [1] program is used for manual segmentation, annotation and transcription.

Currently, our database includes approximately 277 hours of transcribed speech. Statistics of the data in terms of channel and acoustic conditions are given in Table 3.1.

Table 3.1. Amount of Turkish Broadcast News data for various channels and acoustic conditions

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	25.41	9.77	3.38	10.46	42.60	1.66	93.28
NTV	20.34	5.04	3.07	8.92	50.20	2.09	89.66
TRT2	5.57	1.74	0.17	3.32	9.16	0.26	20.22
IE	11.87	0	0	0	0	0	11.87
TRT1	1.16	1.37	0	0.36	2.66	0.14	5.69
VoA	36.46	1.49	7.98	6.21	4.63	0.15	56.92
Total	100.81	19.41	14.60	29.27	109.25	4.30	277.64

3.3. Topic Segmentation & Labeling

3.3.1. Topic Segmentation

136 broadcast news programmes, that are used in SDR experiments, are segmented into news stories such that a news story consists of sentences that refer to the same event (or related events). Story segmentation is very useful for the retrieval of broadcast news programmes. An example news story is shown below:

[1] Transcriber:http://trans.sourceforge.net

Büyük Okyanus'un güneyinde tsunamiye yol açan sekiz şiddetindeki depremde en az on üç kişi öldü. Yerel saatle önceki gün yedi kırkta meydana gelen depremin merkez üssü Solomon Adaları'nın üçyüz elli kilometre kuzeydoğusu olarak açıklandı. Depremden sonra altı virgül yedi şiddetinde artçı sarsıntılar da kaydedildi. Tsunami sonucu çok sayıda köyün suların altında kaldığı en az dört bin kişinin etkilendiği açıklandı. Ölü sayısının artması bekleniyor.

The transcribed audio is manually segmented by topic. It takes about 30 minutes to segment a 1-hour transcribed video by topic. This is a much shorter time compared to utterance segmentation and transcription because reading the transcriptions, instead of listening the audio, makes the process faster. In addition, the transcribed text includes some clues. Speaker changes or long non-speech segments generally indicate a story boundary. As in the case of utterance segmentation and transcription, Transcriber tool is used for topic segmentation and labeling. A snapshot of the process is shown in Figure 3.1.

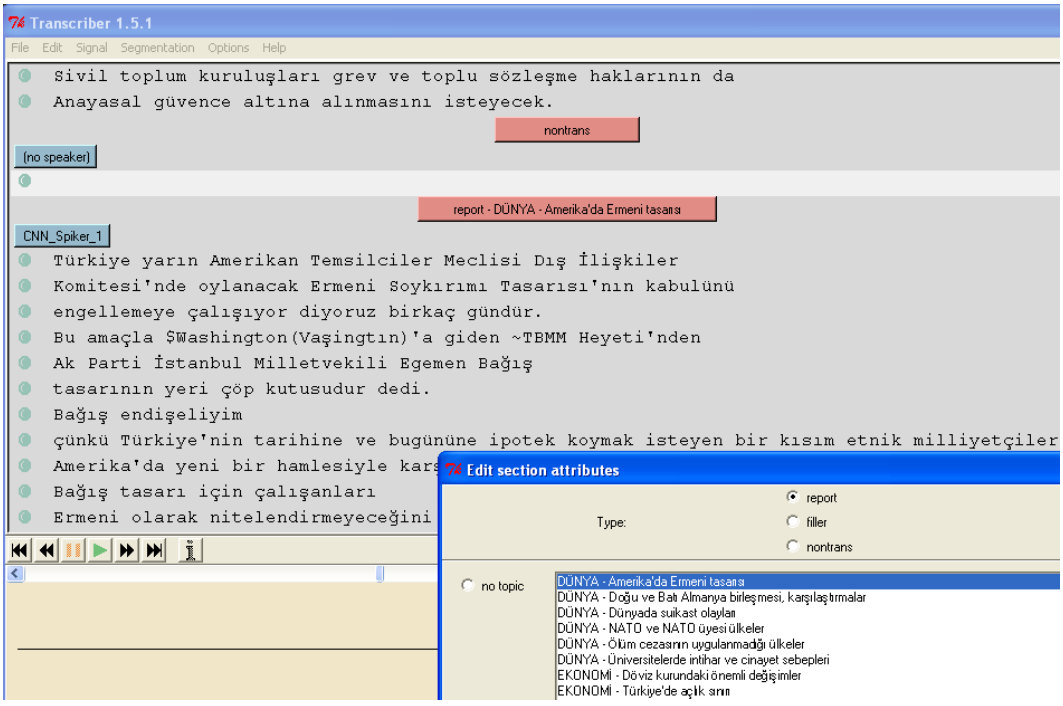

Figure 3.1. Topic segmentation and labeling with Transcriber

In our database, 136 programmes are segmented into a total of 2446 news stories. They have lengths varying from 20 to 3697 words. While some stories consist of only one sentence, some of them include videos, interviews, talks etc. Figure 3.2 shows the histogram of word counts in a news story. The average is 217 words/story.

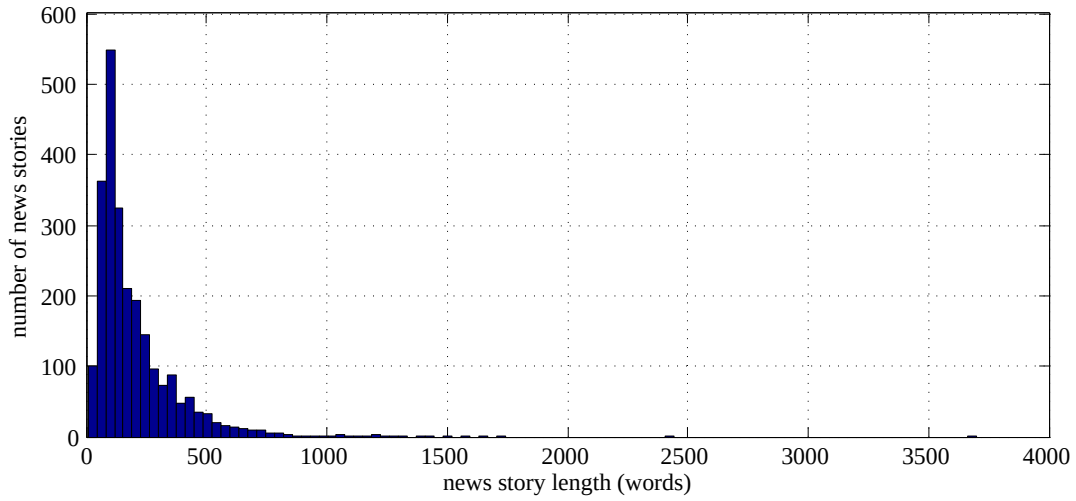

Figure 3.2. Histogram of news story length in words

3.3.2. Topic Labeling

In addition to the story segmentation, each news story is assigned a topic. For this purpose, a TREC-like topic set is created. The 27 topics in the topic set are defined in short and terse forms. A short topic is generally in the structure (and length) of a sentence and a terse topic is the keyword-based counterpart of the short one. An example is presented below, in both forms:

Short Topic:

Türkiye'de ve dünyada son zamanlarda gerçekleşmiş uçak kaçırma, hava korsanlığı vakalarını bul.

Terse Topic:

Uçak kaçırma hava korsanlığı

Complete list of short and terse topics are given in Appendix D. Table 3.2 shows the statistics of both sets.

Table 3.2. Statistics of short and terse topics

	Short Topics	Terse Topics
# of topics	27	27
# of tokens	220	104
# of types	164	95
tokens/query	8.15	3.85

If a news story is not related to any of the topics, then it is labeled as *other*. News summaries, advertisements and weather reports are also labeled but excluded from indexing. Thus, a total of 31 alternatives exist for a news story: 27 topics, 3 excluded topics and "other". Terse forms of the 24 topics and number of related news to each of them are presented in Table 3.3.

Table 3.3. Terse topics and number of related documents in the collection

Topic	# of news
Amerika'da Ermeni tasarısı	32
Doğu Batı Almanya birleşmesi	1
Dünyada suikast olayları	13
NATO ve NATO üyesi ülkeler	11
Ölüm cezası kaldırılması	5
Üniversitelerde intihar cinayet	7
Türkiye'de enflasyon	10
Döviz kurunda değişimler	8
Türkiye'de açlık sınırı	1
Yaz sıcakları önlem	5
En az on ölü doğal afet	14
İstanbul depremi önlemleri	5
İstanbul'da susuzluk tedbirleri	12
Küresel ısınmanın etkileri	34
Türk sineması	2
İşsizlik eğitim ilişkisi	1
Uçak kaçırma hava korsanlığı	2
Kalp hastalıkları	3
Türkiye'de kuş gribi	1
Kanser önleyici gıdalar	1
Türk Kürt Abdullah Gül	4
Seçim Ak Parti yürüyüş	23
Tayyip Erdoğan Amerika ziyaret	13
Türkiye Avrupa Birliği'ne adaylığı	42
Türkiye'de ordu siyaset ilişkileri	26
Doping olayları	2
Türkiye basketbol karşılaşmaları	8

4. SPOKEN TERM DETECTION

The aim of Spoken Term Detection is to retrieve the occurrence time and duration of a query along with a detection score. In this chapter we describe our baseline STD system and give the experimental results. An application of STD is also presented at the end of the chapter.

4.1. System Architecture

The overall STD system consists of 3 main components: automatic speech recognition, indexing and retrieval. ASR and indexing are offline components. Namely, the index is built before actual queries are seen. The online part, retrieval engine, is activated after the user enters the query. The block diagram of the STD system is shown in Figure 4.1.

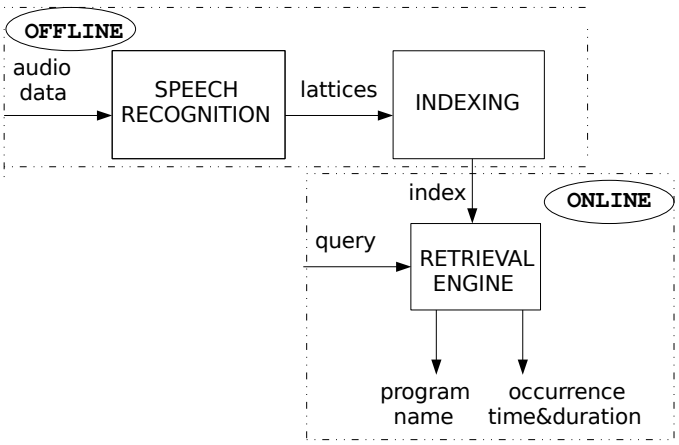

Figure 4.1. Block diagram of the spoken term detection system

4.1.1. ASR

In our STD system, ASR is used to convert the audio information to a textual representation in terms of weighted finite state automata (WFSA). In this work, a traditional HMM based LVCSR system is used.

Our MFCC feature vector consists of 12 cepstrum coefficients, energy components and first and second order derivatives; namely a total of 39 components. Features are extracted using the HTK front end [43].

The acoustic models consist of decision tree state clustered triphones. The output distributions are Gaussian Mixture Models (GMM); each HMM state has a GMM with 11 components (except silence, which has 23 components). The recognition networks and the output hypotheses (one-best and/or alternative hypotheses) are represented as weighted automata. Acoustic model is trained with AT&T's AMTools. The details of the acoustic model can be found in [4].

Language models are word and subword (morph, G-SE, S-SE) based n-grams. Word based model is considered as baseline. For all units, modified Kneser-Ney discounting is used as the smoothing method. Language model is trained with SRILM Toolkit [44].

4.1.2. Indexing

Index is the key component of an IR system, which stores the required information for efficient retrieval. Therefore, an STD index should store the occurrences of terms. In our STD system, weighted automata indexation is employed. Weighted automata indexation is a general framework for efficient retrieval of uncertain inputs [45]. In our case, alternative ASR hypotheses, together with their probabilities, are represented as weighted automata. In the ASR output, all possible substrings are extracted for each utterance and combined via automata union. Resulting index is a transducer, where inputs are the original labels of the automata and outputs are the utterance numbers and expected counts, which can be expressed formally as:

$$EC(l) = \sum_{\pi \in L} p(\pi) C(l|\pi) \tag{4.1}$$

where l is an arc label and π is a path in lattice L. $p(\pi)$ is the posteriror probability of path π and $C(l|\pi)$ is the count of label l in path π.

WFSA indexing enables searching for more complex patterns (phrases, regular expressions) and results in optimal search complexity - linear in the length of the input string.

Below, a simple example of index construction is given. In fact, the input to the indexing algorithm is a weighted automata over the log semiring and all operations are performed over log semiring. However, this example illustrates the process over the real semiring and one-best hypothesis for simplicity.

Let the database consists of two utterances and a, b, c, d represent different words:

$U_1 = aba$
$U_2 = acd$

Recall that ASR output is in terms of finite state automata. The symbolic representation of each utterance is shown in Figure 4.2. The weights next to the labels can be considered as confidence scores, used because of the uncertainty of ASR output. The utterances are represented as confusion networks (see Section 4.2.1.2) and only the solid paths will be indexed for convenience.

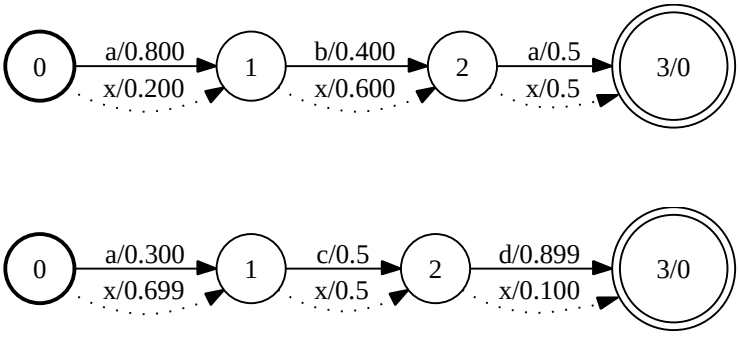

Figure 4.2. The two utterances, to be indexed.

Further steps are going to be explained on the first utterance. Same procedure is going to be applied on the second one.

In *factor selection* part, the utterance is processed to extract its all substrings. First, the fsm is converted to a transducer with epsilon outputs as depicted in Figure 4.3.

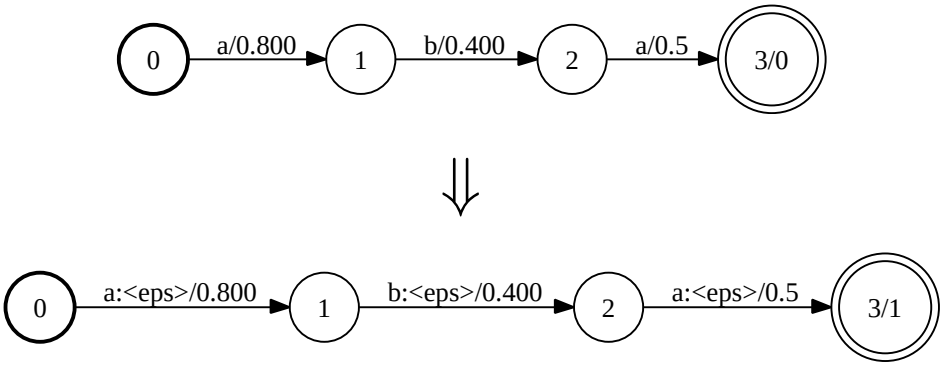

Figure 4.3. Conversion to a transducer with epsilon outputs

Next, a new state is created and assigned as the initial state. New transitions are created from the initial state to each of the previous states. Note that these transitions have "1" as the arc weight, which are the forward probabilities of the previous states.

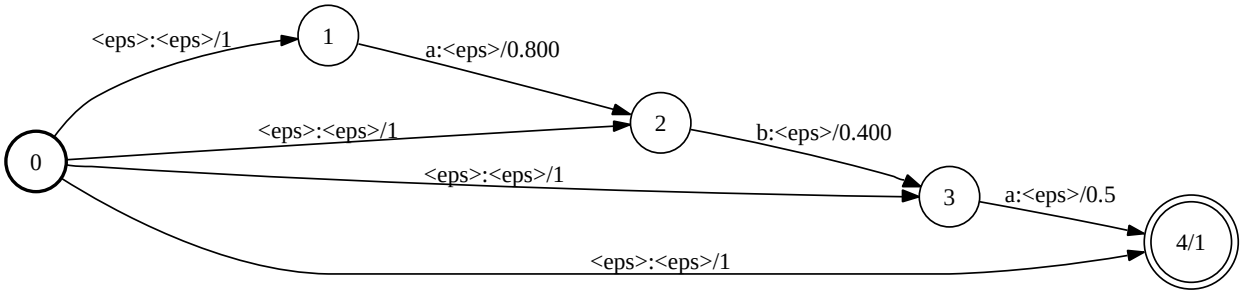

Figure 4.4. Factor selection: a new initial state is added

Similarly, a new state is created and assigned as the final state. New transitions are created from each of the previous states to the new final state. These transitions have the utterance number ("1" in our case, since we are processing the first utterance) as the output.

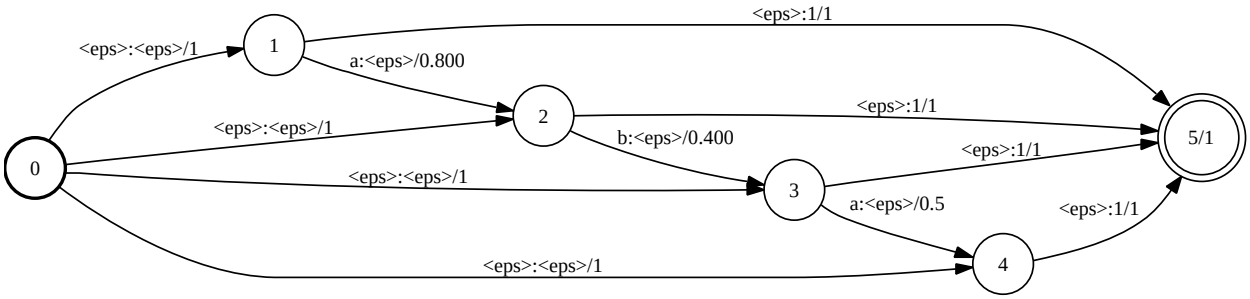

Figure 4.5. Factor selection: a new final state is added

In the *optimization* part, first weighted epsilon removal is applied. The output is shown in Figure 4.6. Next, the fsm is determinized and minimized, which results in the fsm depicted in Figure 4.7.

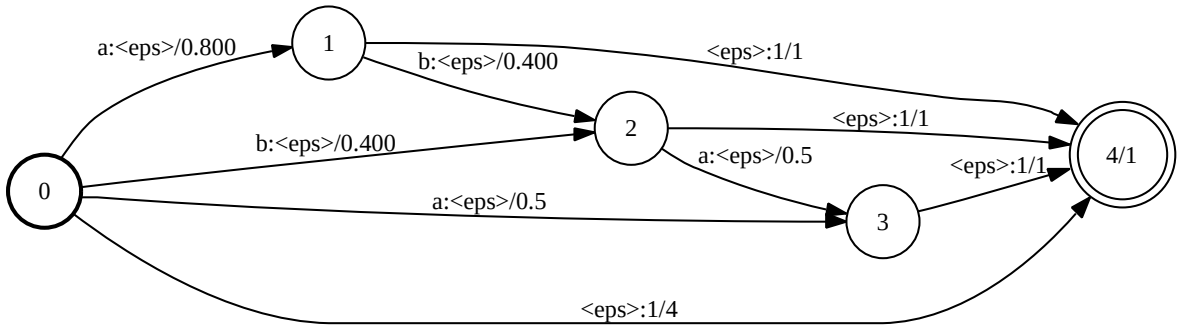

Figure 4.6. Optimization: epsilon removal

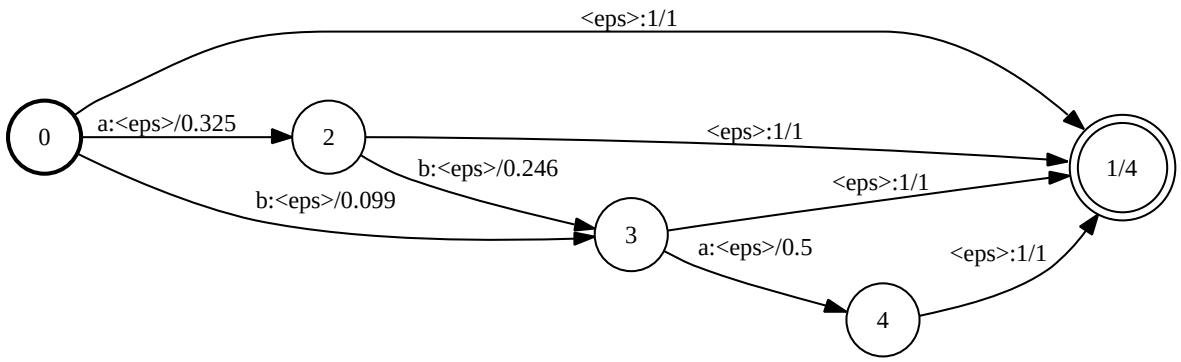

Figure 4.7. Optimization: determinization and minimization

The factor selection and optimization operations are performed for each utterance and all of the processed utterances are combined via automata union. In our case, we

applied the same procedure on utterance 2 and take the union of the two processed utterances. The resulting index is shown in Figure 4.8.

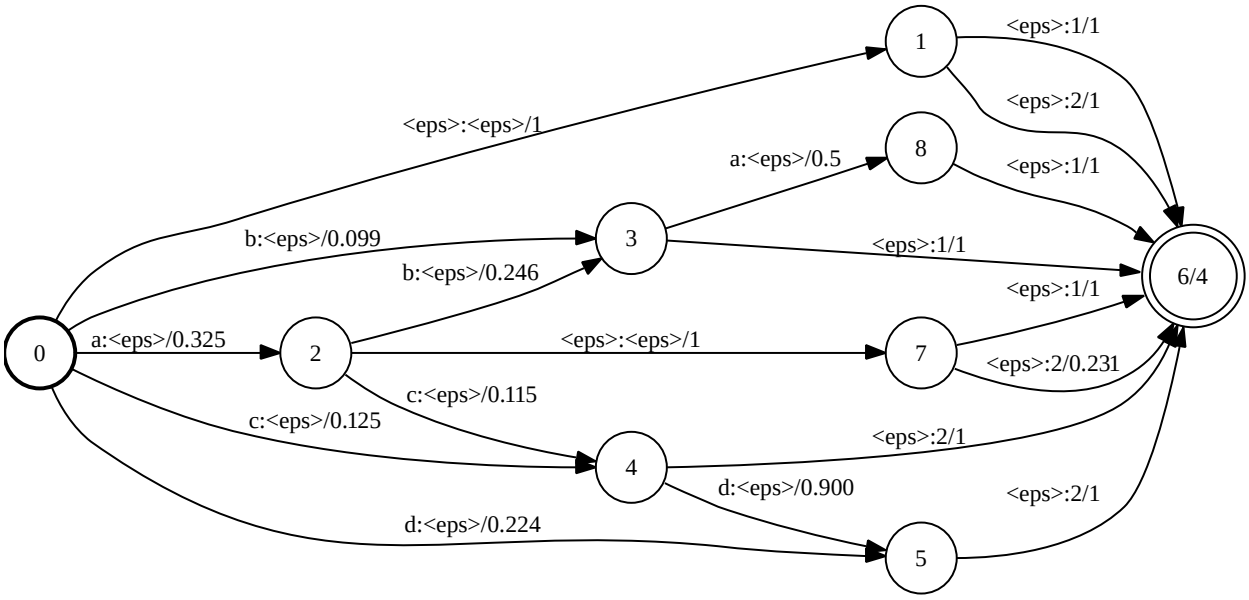

Figure 4.8. The index of utterances *aba* and *acd*

4.1.3. Retrieval

Retrieval is the online component of the system. The queries presented to the system are also represented as finite state automata, and the search is performed by composing these automata with the index transducer. The output contains the list of all utterance numbers where the query appears and the corresponding expected counts. The utterances are ranked using the expected counts, and those exceeding a threshold are selected. It is important to note that, by varying the threshold on the expected count, different operating points can be obtained. After obtaining the utterance indices, we apply forced alignment to identify the starting time and duration of each term [36].

Below, retrieval process is illustrated on the index in Figure 4.8. Recall that the index in Figure 4.8 includes two utterances with the corresponding probabilities: a(0.8)b(0.4)a(0.5) and a(0.3)c(0.5)d(0.9). Let's search for the query word "a". Representation of the query as a finite state automaton is shown in Figure 4.9.

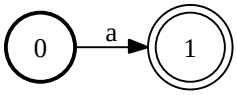

Figure 4.9. FSM representation of the query "a"

For the search operation, query is composed with the index via fsm composition. The output fsm, shown in Figure 4.10, includes two paths.

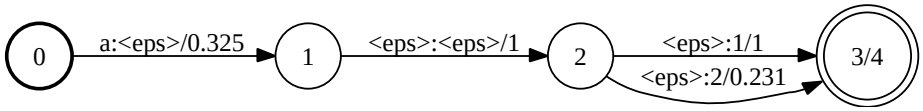

Figure 4.10. Composition output of the query "a" and the index

On the first path:

- input: 'a' (query)
- output: '1' (utterance number which query appears in)
- weight: $0.325 * 1 * 1 * 4 = 1.3$ (expected count of query 'a' in the first utterance)

On the second path:

- input: 'a' (query)
- output: '2' (utterance number which query appears in)
- weight: $0.325 * 1 * 0.231 * 4 = 0.3$ (expected count of query 'a' in the second utterance)

Now let's search for the phrase 'ab' in the same index. The fsm representation is shown in Figure 4.11. The composition output, presented in Figure 4.12, contains only one path.

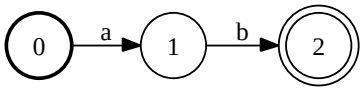

Figure 4.11. FSM representation of the query "ab"

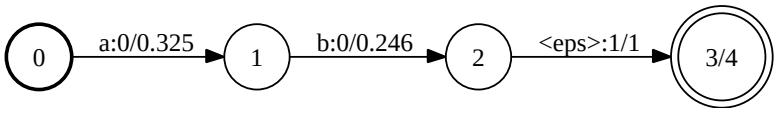

Figure 4.12. Composition output of the query "ab" and the index

On the path:

- input: 'ab' (query)
- output: '1' (utterance number which 'ab' appears in)
- weight: $0.325 * 0.246 * 1 * 4 = 0.32$ (expected count of query 'ab' in the first utterance $0.32 = 0.8 * 0.4$)

4.2. Improvements over the Baseline

4.2.1. Indexing Alternative Hypotheses

4.2.1.1. Lattices. The mechanism of an ASR system requires searching through a network of all possible word sequences. One-best output is obtained by finding the most likely hypothesis. Alternative likely hypotheses can be represented using a lattice. Indexing the whole lattice may save a true (but not best, because of recognition errors) hypothesis from pruning. Thus, indexing the alternative ASR hypotheses makes the STD system more robust to recognition errors. To illustrate, lattice output of a recognized utterance is shown in Figure 4.13.

Use of lattices introduces more than one hypothesis for the same time interval, with different probabilities. Indexation estimates the expected count using these path probabilities. By setting a threshold on the expected count, different precision-recall points can be obtained which results in a curve. On the other hand, one-best hypothesis can be represented with only one point. Having a curve allows choosing an operating point by varying the threshold. Use of a higher threshold improves precision but recall falls. Conversely, a lower threshold value causes less probable documents to be retrieved. This increases recall but decreases precision. The opportunity of choosing the operating point is a great advantage. Depending on the application, it may be

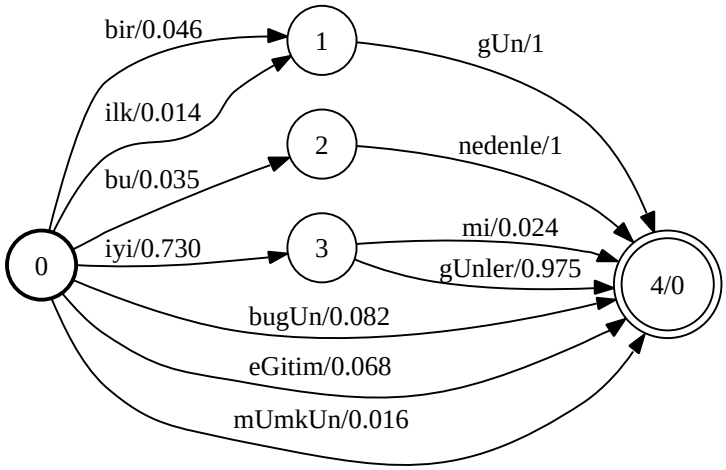

Figure 4.13. An example lattice

desirable to retrieve all of the related documents or only the most probable ones.

4.2.1.2. Confusion Networks. Another way of representing alternative ASR hypotheses is using *confusion networks* (or *sausages*). In a confusion network (CN), all of the word hypotheses are aligned and the ones that are at the same position are grouped into *confusion sets*. Each word in a CN is associated with its a posteriori probability.

A sentence hypothesis is obtained by concatenating the word hypotheses selected from each confusion set. The one-best is the sequence of words, each of which has the highest posterior probability in its set. Note that one-best path of a lattice depends on the probability of the whole sentence. In other words, lattice one-best is the most probable sentence, whereas CN-best is the sequence of most probable words. Therefore, lattices are better in reducing sentence error rate (SER) and CNs are better in reducing WER. However it is important to note that CNs result in extra word sequences that are not represented by the lattice.

A lattice can be transformed into a CN. An example is depicted in Figure 4.14, which is generated from the lattice in Figure 4.13. Note that, the path *bir eğitim* can be extracted from the CN but it is not allowed by the lattice. In our experiments, lattices are transformed into CNs using the SRILM Toolkit [44].

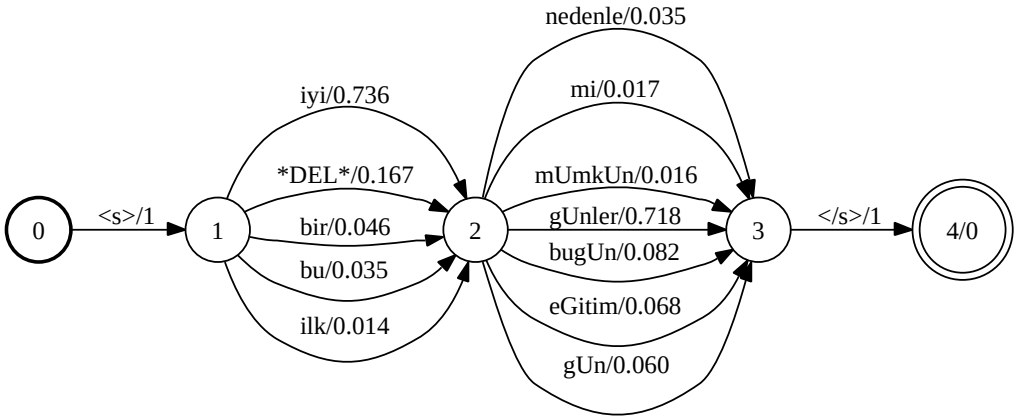

Figure 4.14. An example confusion network

4.2.2. Use of Subword Units in Language Modeling

As explained in section 2.1.2.1, OOV words can not be recognized by ASR and located by the STD system. OOV problem is more severe for Turkish because of its agglutinative structure. We experimented with the following subwords as the language modeling/indexing unit, to deal with the OOV problem. We index the language modeling units directly, except phones. The queries are segmented into the proper units prior to search. All subword based vocabularies are restricted to 50k units in ASR, as in the case of word vocabulary.

4.2.2.1. Phones. Phone indexing was shown to be successful in [34] for English. Since Turkish is an almost phonetic language, our acoustic models are based on graphemes (or letters), instead of phonemes. That's why we directly index the graphemes. Since phone recognition is less accurate than word recognition, we preferred using words/morphs in recognition and transducing the words/morphs to graphemes before indexing.

4.2.2.2. Grammatical Stem-Endings. We use a morphological parser that is developed in Boğaziçi University, which is based on the two-level morphology. The morphophonemic rules and lexicon have been adapted from PC-Kimmo implementation of Kemal Oflazer [46]. In Turkish, a single word may have multiple interpretations and parses due to the complex morphology. Although disambiguation is helpful to find the correct

parse, currently the parser does not apply disambiguation but chooses the parse with the minimum number of morphemes. An example parse of word *çocukları* is given below [4].

```
Çocuk[Noun]+lAr[A3pl]+SH[P3pl]+[Nom] (onların çocukları)
Çocuk[Noun]+lAr[A3pl]+SH[P3sg]+[Nom] (onun çocukları)
Çocuk[Noun]+lAr[A3pl]+[Pnon]+YH[Acc] (çocukları [gördüm])
Çocuk[Noun]+[A3sg]+lArH[P3pl]+[Nom] (onların çocuğu)
```

The output of the parser is simplified by removing the part-of-speech tags and the morphological features. First morpheme is assigned as the stem and the other morphemes are grouped in their surface form to constitute the ending. In our G-SE based 50k lexicon, 41363 of the units are stems and 8637 of the units are endings. Coverage of this lexicon on the test set is about 99%.

4.2.2.3. Morphs. The segmentation software, Morfessor, is used to obtain the statistical subword units [11]. Only the words that are seen more than 20 times are used for morph generation. No word boundary markers are used but a "+" symbol is attached to the suffixes (like G-SEs). The overall text corpus is segmented into approximately 142000 stems (first morph) and 43000 suffixes (other morphs). The 50k unit, morph-based lexicon consists of 46645 stems and 3355 suffixes. The coverage of the morph-based lexicon is about 95%.

4.2.2.4. Statistical Stem-Endings. To generate S-SEs, output of the morph segmentation is processed as explained in Section 2.1.2.1. The resulting lexicon consists of 41651 stems and 8349 endings and covers about 99% of the test set. Note that these statistics are very similar to those of G-SE's, however the number of stems and coverage is higher with a morph-based vocabulary.

To illustrate the difference between these units, various parses of the word se-

quence *dünya kupası finalinde* are given below:

Word : dünya kupası finalinde
Phone: d ü n y a k u p a s ı f i n a l i n d e
G-SE : dünya kupa +sı final +inde
Morph: dün +ya kupası final +in +de
S-SE : dün +ya kupası final +inde

4.2.3. Cascades

It is possible to employ both word and subword indexes for retrieval. Several methods are proposed for combination such as interpolation [33], WFSA composition [37], utilization of time-stamps [35] and cascading [34]. We experiment with the two cascading strategies defined in [34]:

1. Vocabulary Cascade: IV queries are searched in word-based index and OOV queries are searched in subword based index.
2. Search Cascade: First the query is composed with the word index. If no occurrences are found, it is searched in the subword index.

4.2.4. Term Specific Thresholding

We have already mentioned in Section 4.2.1.1 that, it is possible to have different precision-recall points by setting a threshold on the expected count. Instead of applying the same detection threshold on all queries, it is possible to determine an optimum threshold for each query. In [31] and [30], term thresholding is shown to be successful. The thresholds are determined to maximize the ATWV metric.

To calculate the optimum thresholds, a formula is proposed in [31]:

$$th(q_k) = \frac{R(q_k)}{\frac{T_{speech}}{\beta} + \frac{\beta-1}{\beta} R(q_k)} \tag{4.2}$$

The exact value of $R(q_k)$ is not known but approximating this number by the expected number of occurrences of q_k in the test corpus is reasonable [36].

4.3. Experiments - First Set

4.3.1. Experimental Setup

4.3.1.1. Evaluation Metrics. The metric for ASR evaluation is the standard Word Error Rate (WER) metric. Recognition performances of of various systems are compared in WER and significance level is measured by NIST MAPSSWE significance test. Retrieval part is evaluated via various metrics. The first two, Precision-Recall rates and F-measure are traditional information retrieval metrics. We also used the ATWV metric, defined by NIST. To test the statistical significance of retrieval part, paired t-tests are conducted on maxF results with MATLAB.

4.3.1.2. Training Corpora. For the first set of experiments, the acoustic model is trained on approximately 100 hours of speech from our Turkish Broadcast News recordings. Amount of acoustic data in terms of channel and acoustic conditions is presented in table 4.1.

Table 4.1. Amount of training data for various channels and acoustic conditions

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	8.6	5.0	0.8	3.4	11.6	0.5	29.9
NTV	7.8	2.3	0.9	2.4	17.2	0.9	31.5
TRT2	4.4	1.3	0.1	2.5	6.9	0.2	15.4
HI	10.1	0	0	0	0	0	10.1
TRT1	1.6	1.4	0	0.3	2.7	0.1	6.1
VoA	4.7	0.2	1.0	0.8	0.6	0.3	7.6
Total	37.2	10.2	2.8	9.4	39.0	2.0	100.6

The overall language model is a combination of two models: general and specific. General language model is trained on a large corpus of 96M words, which is collected

from web (mostly newspapers). Specific language model is built using the manual transcriptions of the acoustic data. These transcriptions are more concordant with the broadcast news recognition task, however the number of words is much smaller. As explained in section 2.1.2, both amount of data and domain are important issues in language modeling. To utilize the advantages of both general and specific models, they are interpolated. Linear interpolation is a commonly used method for language model adaptation. It is formulated as follows:

$$P(w_3|w_2,w_1) = \lambda P_G(w_3|w_2,w_1) + (1-\lambda)P_S(w_3|w_2,w_1) \qquad 0 \leq \lambda \leq 1$$

In Equation 4.3, P_G stands for general model and P_S stands for specific model. The interpolation constant λ is optimized empirically to 0.5.

4.3.1.3. Test Corpora. Two types of test corpora are used for the first set of experiments. First one, the Turkish Broadcast News (BN) Corpus, includes about 4 hours of speech in various acoustic conditions. Table 4.2 analyzes the amount of BN test data in terms of channel and acoustic conditions. Second one is the Turkish Broadcast News for the Hearing Impaired (HI) Corpus, which includes about 10 hours of clean and clearly articulated speech (f0). There is no overlap in dates between the training corpus and the test corpora. Both of the test corpora are used to analyze the improvement provided by lattice usage in different acoustic conditions (Using word lattices - Section 4.3.2.1). Other experiments are performed only on the second corpus.

Table 4.2. Amount of BN test data for various channels and acoustic conditions

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	0.25	0.12	0	0.07	0.25	0	0.69
NTV	0.16	0	0.02	0.09	0.34	0.02	0.63
TRT2	0.17	0	0.04	0.01	0.45	0.01	0.68
IE	0.95	0	0	0	0	0	0.95
VoA	0.49	0.05	0.16	0.08	0.05	0	0.83
Total	2.02	0.17	0.22	0.25	1.09	0.03	3.78

The WER and OOV statistics of both corpora, with a vocabulary of 50k words, are given in table 4.3. Note that, WER of BN corpus is much higher than HI corpus. Typewise (each type of word is considered once; previously seen words are ignored) OOV rates are over 20% and tokenwise (each word is taken into account, independent of the type) OOV rates are also quite high for both test corpora.

Table 4.3. WER and OOV rates for different corpora computed using the word based language model with a vocabulary of 50k

		OOV rate		# of words	
corpus	WER	type	token	type	token
BN	40.3%	21.6%	7.9%	8932	35314
HI	23.2%	23.4%	6.3%	12804	49903

ASR performance of word and morph based language models are compared on HI corpus. Best paths are extracted both from the CN and lattice. As can be seen in Table 4.4, morph-based model provides a significant improvement ($p < 0.001$) and best hypothesis of the CN is superior than that of the lattice ($p < 0.001$).

Table 4.4. WER of different methods and LM units on HI Corpus using 50k vocabularies

unit	one-best	CN-best
word	23.2%	22.1%
morph	20.4%	20.0 %

4.3.1.4. Queries. The query set of each corpus consists of all the words seen in manual transcriptions, excluding foreign names and acronyms. These sets contain only single word queries, they do not contain any phrases. That's why, for each query set, the percentage of OOV queries in Table 4.5 is approximately equal to its typewise OOV rate in Table 4.3. With the morph based vocabulary, both of the query lists are completely covered.

Table 4.5. Number of queries and OOV rates of both query sets

	OOV rate	# of words
BN	21.0%	8719
HI	22.6%	9635

4.3.2. Experimental Results

4.3.2.1. Using Word Lattices. We experimented with one-best and lattice indexes to investigate the effect of lattice based search. The resulting precision-recall graph for HI corpus is shown in Figure 4.15. Note that, lattice approach corresponds to a curve in the plot, while one-best approach seems almost like a point. The curve of the one-best approach can be noticed merely by zooming in the plot, as shown in Figure 4.16. Therefore we can conclude that, the lattice based index provides a wide flexibility of choosing an operating point. On the other hand, only an extremely small operating region can be obtained by indexing the one-best hypothesis.

Another point to note is that, the one-best curve is below the lattice curve. Namely, it is always possible to obtain a better point using the lattice-based index. This strengthens the superiority of lattice usage.

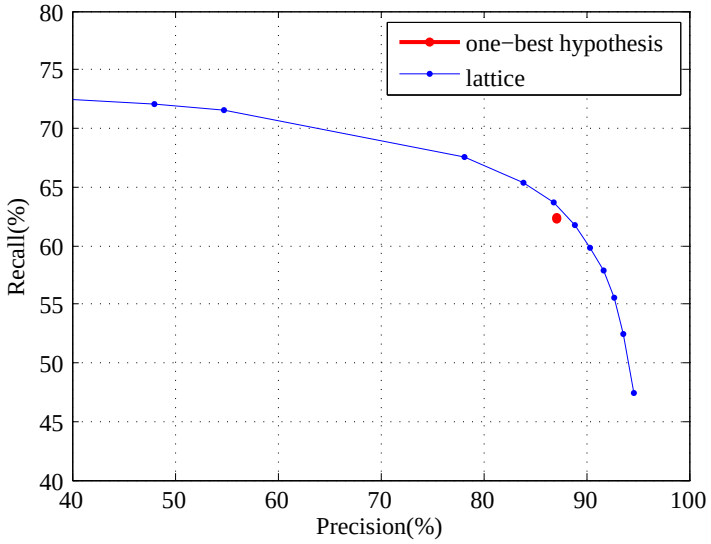

Figure 4.15. Precision-recall graph for one-best and lattice indexing approaches on HI corpus

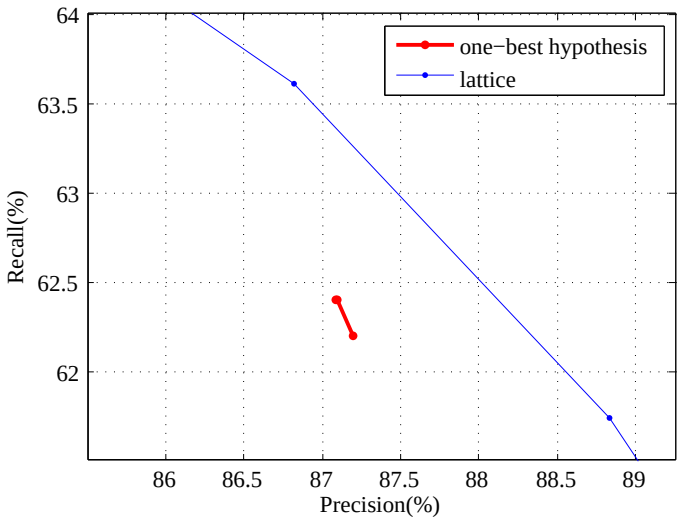

Figure 4.16. Precision-recall graph for one-best and lattice indexing approaches on HI corpus - zoomed version

Next, we investigate the effect of lattice based search on BN corpus and compare with HI corpus. We also compare these with indexing the one-best hypothesis obtained using confusion networks on the HI corpus. As can be seen in Table 4.6, lattice-based search improves the performance for both corpora. However, the improvement is higher in BN corpus, which has a higher WER. The one-best obtained from CNs gives only a very slight improvement over the baseline one-best.

Although the improvement provided by CNs is very small, statistical testing on the HI corpus shows that it is significant ($p = 0.01$). Lattices also provide a significant gain over the one-best ($p \approx 0$)[2] . Note that the p-value for one-best vs lattice comparison is much lower, which can be considered as a stronger evidence of difference between methods. Superiority of CN-best, on the other hand, is relatively more uncertain.

4.3.2.2. Using Morph Lattices and Cascades. Subword based language models are used in ASR and indexing to reduce the OOV rate. In this experiment, morph lattices are employed in the system and compared with word lattices. The morph-based index pro-

[2]This notation is used when $p < 0.001$

Table 4.6. Maximum precision, recall, F-measure and MTWV values for one-best, CN-best, and lattice on BN and HI corpora. "Improvement" is the absolute difference between one-best and lattice.

	maxP (%)	maxR (%)	maxF (%)	MTWV (%)
one-best	82.2	48.0	60.5	46.6
lattice	94.3	63.2	62.9	51.8
improvement	12.1	15.2	2.4	5.2

(a) BN corpus

	maxP (%)	maxR (%)	maxF (%)	MTWV (%)
CN-best	87.3	62.3	72.7	60.9
one-best	86.9	62.3	72.4	60.8
lattice	94.6	72.7	73.5	64.5
improvement	7.7	10.4	1.1	3.7

(b) HI corpus

vides much better maxF and ATWV scores as can be seen in Table 4.7. Statistical tests support the superiority of maxF scores with a significance level of $p \approx 0$. Figure 4.17 shows the precision-recall graphs of both word and morph based approaches. Morph based recognition and indexing increases the recall significantly, however maximum precision attainable is lower than the word index.

Having different characteristics, these two approaches can be combined via cascading strategies mentioned in Section 4.2.3. The results, presented in Figure 4.18 and Table 4.7 show that both cascading methods outperform the individual indexes, which are also shown to be statistically significant ($p \approx 0$). Vocabulary cascade is better in the very high precision region and search cascade is better in the remaining region. In Table 4.7, we use F-measure and MTWV metrics to compare the cascading strategies. The search cascade strategy performs slightly better than the vocabulary-cascade in terms of maxF and MTWV. The difference is statistically significant with $p \approx 0$.

It is interesting to note that the improvement in retrieval performance is much

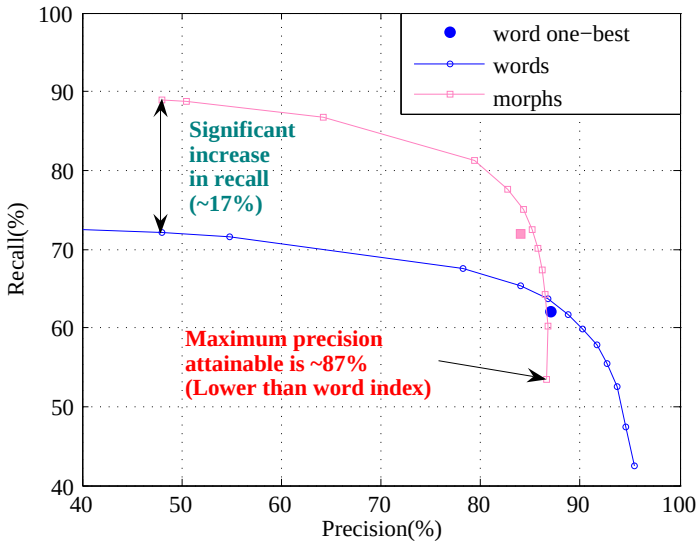

Figure 4.17. Comparison of word and morph based indexing on HI corpus. Solid single markers indicate the performance of one-best approaches.

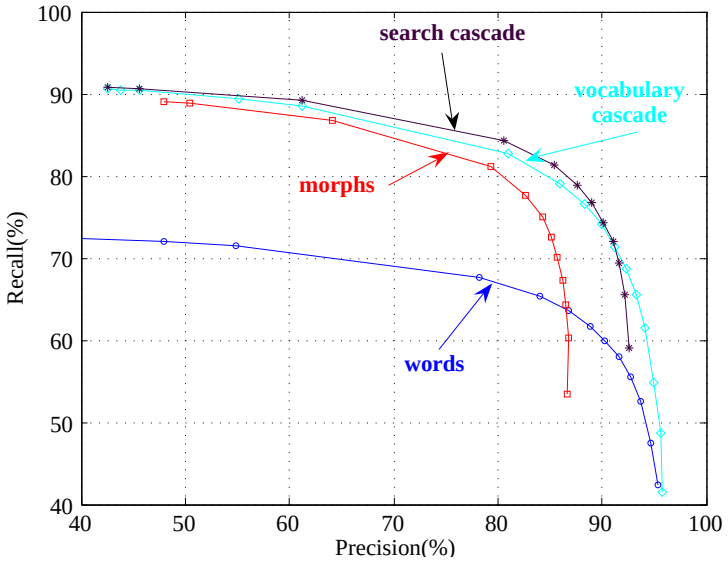

Figure 4.18. Comparison of word, morph and hybrid indexing strategies on HI corpus.

more impressive than the improvement in WER when morph-based language models and indexes are used instead of word-based language models and indexes. The reason can be explained as follows. In retrieval, typewise OOV rates are in effect because each word occurs only once in our query set. However in ASR, tokenwise OOV rates are in effect. Since typewise OOV rate is greater, alleviating the OOV effect ensures a higher improvement in retrieval.

Table 4.7. Performance of various methods in maximum F-measure and MTWV (VC:vocabulary cascade, SC:search cascade)

	word	morph	VC	SC
maxF (%)	73.5	80.3	82.4	83.3
MTWV (%)	64.5	75.8	79.5	81.1

4.3.2.3. Using Phones as Subword Units. Since phone recognition performance is worse than word and morph recognition, phones are used only in indexing. Two different phone indexes are built: one by converting words to phones and another by converting morphs to phones. The "phone" curve in Figure 4.19 is obtained by converting morphs to phones. As can be seen in Figure 4.19, the individual phone index improves recall over the word index and even outperforms the individual morph index slightly at low precision points. However, as precision gets higher, recall degrades dramatically. In other words, although the maximum recall attainable by using phone-based indexing is slightly larger, the maximum precision attainable is much lower. Thus the individual phone index does not contribute much to retrieval performance.

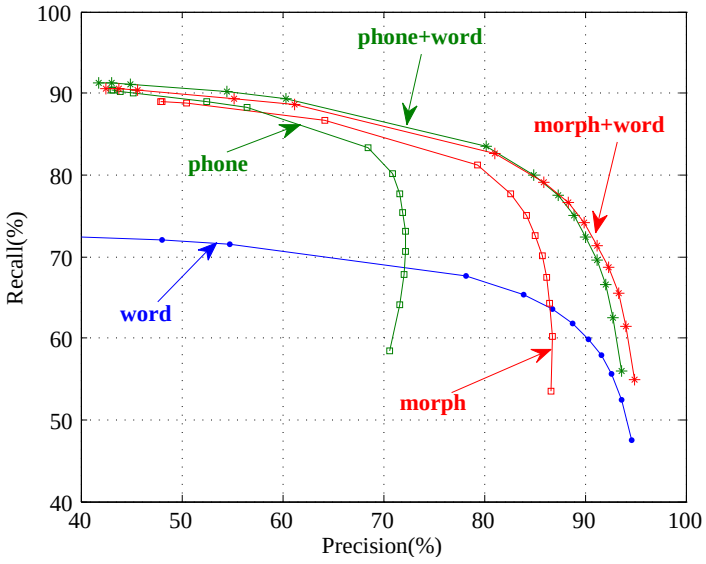

Figure 4.19. Comparison of word, morph, phone and hybrid indexing strategies on HI corpus.

Cascading the word and phone indexes, performance gets better but not better than the word morph cascade. Moreover, the word-phone cascade has a significantly worse performance in maxF than the word-morph cascade ($p \approx 0$). From these results it can be concluded that, unlike English, phone indexing is not so beneficial for Turkish. This might be due to the fact that Turkish is almost a phonetic language and we base our acoustic models on graphemes instead of phonemes. It could also be argued that the gain from phonetic indexing in the case of English is due to homophones.

4.3.2.4. Using Term Specific Thresholds. In this experiment, we apply term specific thresholds in detection and compare with global thresholding. Applying a specific threshold for each term, a single point can be obtained, instead of a curve. By changing the β in Equation 4.2, it is possible to obtain different operating points which can be represented as a curve. For the NIST 2006 STD Evaluation the β value was taken to be approximately 1000. Incrementing β, the penalty of false alarms is increased and higher precision points can be obtained. As presented in Figure 4.20 and Table 4.8, term thresholding introduces a gain to search cascade, which is statistically significant with $p \approx 0$.

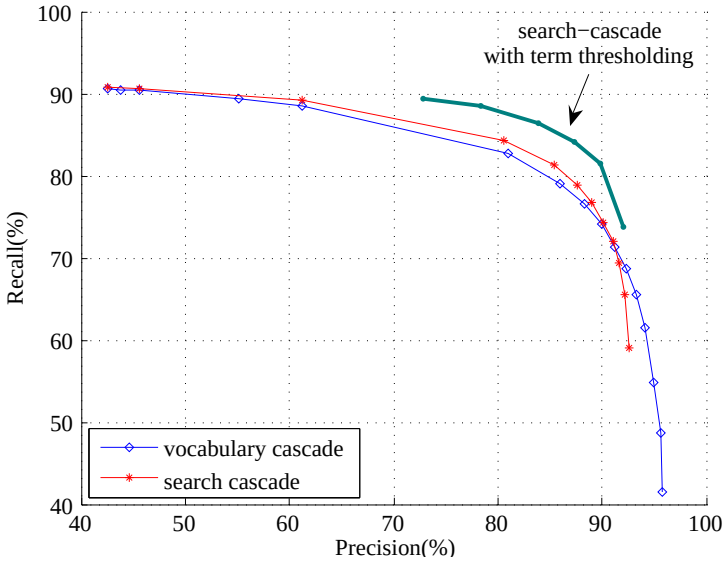

Figure 4.20. Precision-recall curves of word-morph cascades and search cascade with term thresholding

Table 4.8. Performance of various methods in maximum F-measure and MTWV (VC:vocabulary cascade, SC:search cascade, TTh: term thresholding)

	VC	SC	SC+TTh
maxF (%)	82.4	83.3	85.6
MTWV (%)	79.5	81.1	85.7

Term thresholding is also applied to the individual word and morph based indexes. Results are presented using P_{miss} vs P_{FA} curves in Figure 4.21.

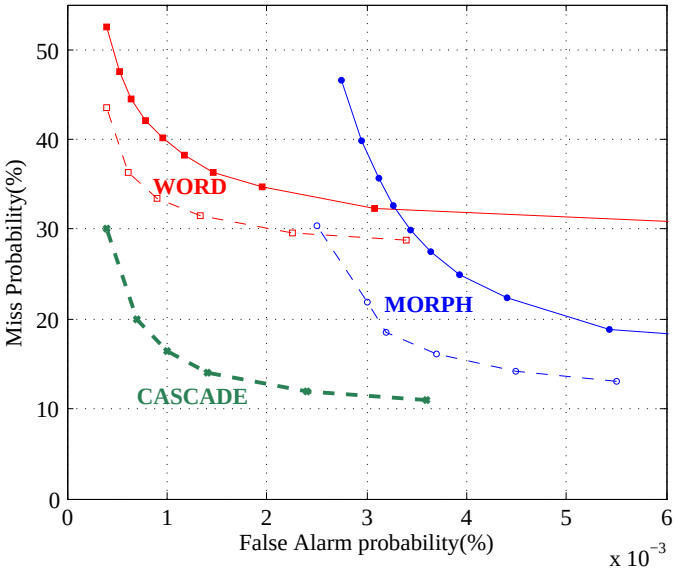

Figure 4.21. P_{miss}-P_{FA} curves of term specific thresholding and global thresholding on HI corpus. Solid lines represent using a global threshold while dashed lines represent using optimal term-specific thresholds.

4.4. Experiments - Second Set

In this second set of experiments, we evaluate our system using STDEval, an evaluation tool developed by NIST.

4.4.1. Experimental Setup

4.4.1.1. Training Corpora. For the second set of experiments, we extended the acoustic training set from 100 hours to $\sim$ 180 hours of speech. Table 4.9 shows the amount of training data for various channels and acoustic conditions.

Table 4.9. Training data analysis for second set of experiments, in terms of channel and acoustic conditions.

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	15.0	7.8	1.9	6.6	22.9	1.0	55.2
NTV	15.5	3.8	2.2	6.8	36.9	1.7	66.9
TRT2	5.1	1.6	0.2	2.7	8.0	0.2	17.8
IE	11.9	0	0	0	0	0	11.9
TRT1	1.2	1.4	0	0.4	2.6	0.1	5.7
VoA	17.0	0.9	4.0	3.0	1.3	0.1	26.3
Total	65.7	15.5	8.3	19.5	71.7	3.1	183.8

The language model is also improved with a new and larger corpus. It is collected from three major news portals in Turkish with a web crawler and contains about 184M words of text. Details about this corpus can be found in [47]. Manual transcriptions of the acoustic data are also used for language model interpolation.

4.4.1.2. Test Corpora. All of the experiments are performed on two different types of test corpora. First one is HI corpus, which has been previously used in the first set of experiments (Section 4.3.1.3). Second one is BN-2 corpus, which is similar to BN corpus of the first set of experiments in channel and acoustic variability but they are

not identical. Table 4.10 shows the amount of BN-2 test data in terms of channel and acoustic conditions. There is no overlap in dates between the training and test data.

Table 4.10. Analysis of BN-2 test data in terms of channel and acoustic conditions

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	0.13	0.02	0	0.12	0.45	0	0.72
NTV	0.18	0.09	0.03	0.01	0.28	0.01	0.69
TRT2	0.18	0.03	0	0.22	0.46	0	0.89
VoA	0.62	0	0.04	0.08	0.07	0	0.81
Total	1.11	0.14	0.07	0.52	1.26	0.01	3.11

WER and OOV statistics of both corpora are shown in Table 4.11. Note that the word error rates are much lower in the second set of experiments. A possible reason is the increase in amount of training data both in acoustic model and language model.

Table 4.11. WER and OOV rates for different corpora

		OOV rate		# of words	
corpus	WER	type	token	type	token
BN-2	31.5%	17.4%	7.2 %	7615	23038
HI	21.7%	21.2%	6.8 %	12804	49903

4.4.1.3. Queries. For the second set of experiments, the query set is composed of 4 subgroups:

- in vocabulary, in reference queries (IV-IRef)
- out of vocabulary, in reference queries (OOV-IRef)
- in vocabulary, out of reference queries (IV-OORef)
- out of vocabulary, out of reference queries (OOV-OORef)

Out of reference (OORef) queries are the ones that are not seen in manual transcriptions. Namely, if the search engine returns a document for an OORef query, it is definitely a false alarm. The reason of using OORef queries is to measure the false alarm rate of the system, when search term is not included in database.

In reference (IRef) queries are selected with the "Term Selection Tool" developed by NIST. This software randomly selects terms based on manual transcriptions. First, all of the term statistics are computed. The most frequent terms are ignored and remaining ones are grouped into s bins, according to their frequencies. Here, s is a user defined parameter. Next, one term is chosen randomly from each group, resulting in s queries. Same process is performed for bigrams, trigrams and fourgrams. We use the default numbers in the original scripts and generate a set of 1082 queries for BN corpus and 1092 queries for HI corpus. This process produces IV-IRef and OOV-IRef queries. (Recall that a query is said to be an OOV query if it includes one or more OOV words.) To increase the OOV rate, we extract the OOV words in the reference transcripts and eliminate the ones that are currently in the OOV-IRef set. A group of queries are selected randomly from the remaining words and added to the OOV-IRef query set. Number of queries in each group is shown in Table 4.12 for both corpora.

Table 4.12. Number of queries in each subgroup. OOV rates are calculated using the 50k vocabulary

	BN-2		HI	
	IV	OOV	IV	OOV
IRef	937	415	922	501
OORef	249	197	199	149
OOV Rates	34%		36%	

Unlike the query set in the first set of experiments, this set includes phrases, foreign words, acronyms and OORef words. In other words, it is a more realistic query set.

4.4.1.4. Evaluation Metrics. The system is evaluated using "Spoken Term Detection Evaluation Toolkit (STDEval)" developed by NIST. STDEval computes system performance in terms of ATWV, using the reference transcripts, query list and search results. Definition of ATWV metric is given in Section 4.3.1.1. Detailed analysis of detection performance is depicted via detection error tradeoff (DET) curves, which are variants of ROC curves. DET curves are plotted with error rates on both axes, P_{miss} vs P_{FA}. A scale is used for both axes, which spreads out the plot and makes it more close to linear [48]. Detailed explanation of NIST STD Evaluation and Toolkit can be found in [24].

4.4.2. Experimental Results

4.4.2.1. Effect of Lattice Pruning Threshold. As concluded in Section 4.3.2.1, lattice usage always improves the system performance. Lattices can be pruned to contain only the paths whose costs (i.e. negative log likelihood) are within a threshold with respect to the best path. The smaller this cost threshold is, the smaller the lattices and the index files are. In this experiment, we investigate the effect of lattice pruning threshold.

Table 4.13 and Figure 4.22 show the comparison of varios pruning thresholds on BN-2 data. As can be seen, MTWV value is not affected much by lattice pruning threshold. So, we skipped repeating this experiment on HI corpus, which will probably introduce a smaller change. (Section 4.3.2.1 has shown that, improvement by lattices is higher in BN corpus)

On the other hand, in low P_{FA} region (i.e. high precision region), higher thresholds demonstrate better performance. Moreover, it is possible to obtain smaller miss probabilities by increasing the pruning threshold. Thus, it might be reasonable to set higher thresholds depending on the application. As presented in Table 4.13, the index size and indexing time is much higher for $th = 8$, that's why we set $th = 4$ for the rest of the experiments.

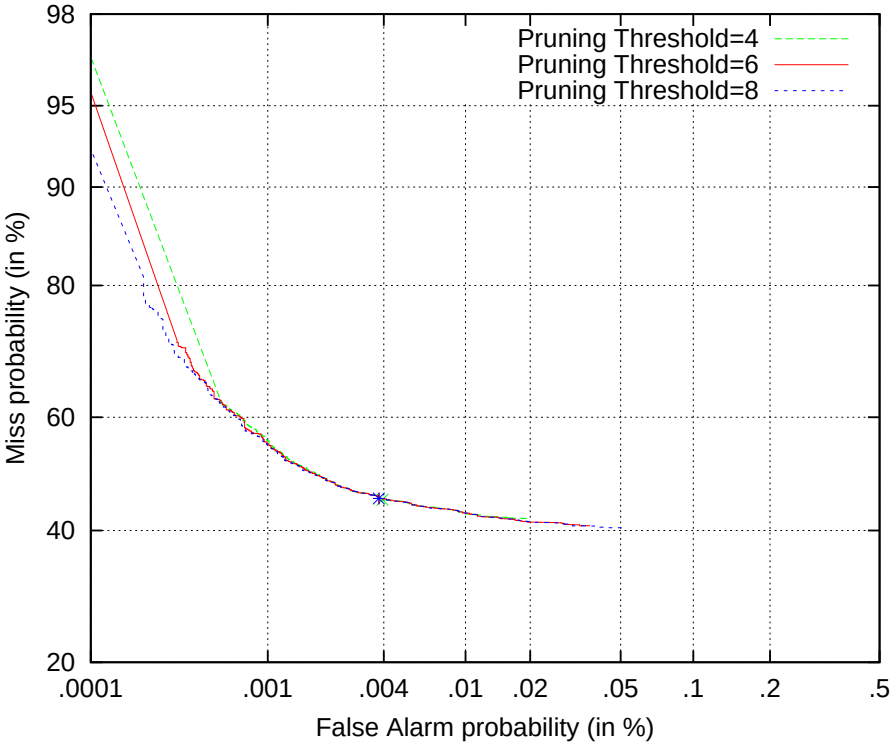

Figure 4.22. Comparison of various lattice pruning thresholds on BN-2 corpus. The bigger marker on each curve indicates the point where MTWV is achieved. (The markers are not distinguishable in this plot since they are very close to each other.)

Table 4.13. Index size, indexing time, MTWV and min P_{miss} values of lattices with various pruning thresholds on BN-2 corpus

	index size	indexing time	MTWV	min P_{miss}
th=4	8.5MB	4 sec.	50.65%	42.09%
th=6	40.8MB	48 sec.	50.55%	40.85%
th=8	268.0MB	38 min.	50.58%	40.48%

4.4.2.2. Effect of Vocabulary Size. To analyze the effect of vocabulary size on STD, 50k and 200k word vocabularies are used in ASR. MTWV results are given in Table 4.14 for both corpora, along with the WER and OOV rates. In the table, OOV-t shows the tokenwise OOV rate of test corpus (more related to WER) and OOV-q is the OOV rate of query list (more related to MTWV). A query is assumed to be OOV if it includes one or more OOV words.

Table 4.14. WER, OOV rates and MTWV for different vocabulary sizes.

	BN-2				HI			
	OOV-t	WER	OOV-q	MTWV	OOV-t	WER	OOV-q	MTWV
50k	7.2%	31.5%	34.6%	50.65%	6.8%	21.7%	36.6%	55.45%
200k	1.9%	27.8%	27.3%	55.25%	1.7%	16.4%	29.0%	61.73%

As expected, decrease in OOV rate directly improves the speech recognition and STD results. Paired t-tests on MTWV point out that 200k vocabulary performs significantly better than 50k vocabulary for both corpora ($p < 0.05$). The improvement in MTWV and DET curves are slightly higher for HI corpus. This might be due to the higher OOV rate of HI query list.

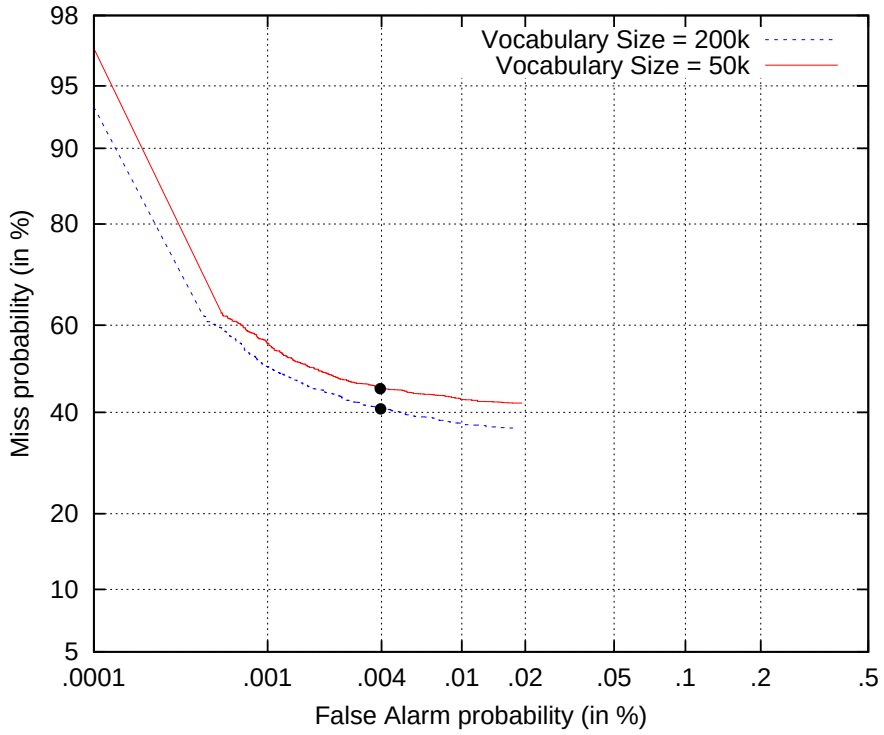

Figure 4.23. DET curves for 50k and 200k word vocabularies on BN-2 corpus. Markers indicate the MTWV point.

4.4.2.3. Use of Subword Units. Recall that, we experimented with morphs as the subword unit in initial experiments. In the second set of experiments, grammatical and statistical stem-ending units are used in recognition and indexing, in addition to morphs.

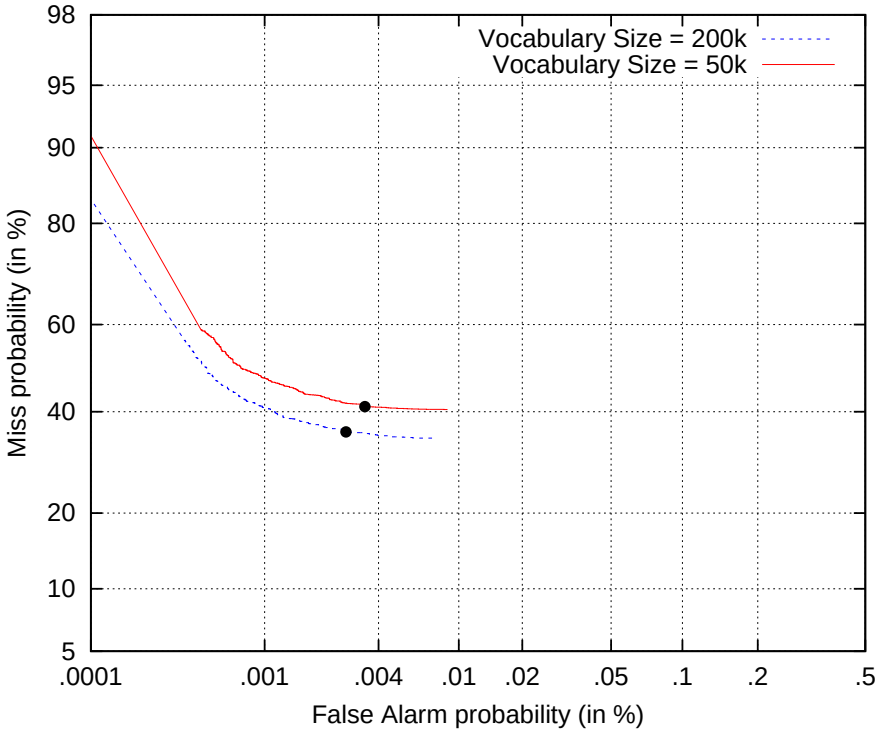

Figure 4.24. DET curves for 50k and 200k word vocabularies on HI corpus. Markers indicate the MTWV point.

OOV rates and recognition results of the subword based language models are given in Table 4.15. To compare, WER and OOV rate of the word-based models are also presented. Note that, the smallest length units, morphs, have the lowest OOV rates. Zero OOV rate can not be achieved with morphs in our case since the lexicon is restricted to 50k units.

The DET curves of word and subword based models are depicted in Figures 4.25 and 4.26 for the two corpora. Stem-ending units, as in the case of morphs, make it possible to obtain smaller miss probabilities but with a cost of increasing false alarm rate. G-SE based index achieves the best performance among the subword units. At the maximum P_{FA} region, morph based index gets slightly better for HI data and becomes equal for BN-2 data. The common behavior is that morphs introduce much smaller miss probabilities with higher false alarm rate. From these observations, we

Table 4.15. WER and OOV rates of various units for both corpora

	BN-2			HI		
	WER	OOV-t	OOV-q	WER	OOV-t	OOV-q
Word (50k)	31.5	7.2%	34.6%	21.7	6.8%	36.6%
Word (200k)	27.8	1.9%	27.3%	16.4	1.7%	29.0%
Morph	26.1	0.6%	7.6%	13.8	0.5%	10.2%
S-SE	25.6	1.0%	11.8%	14.1	0.8%	13.1%
G-SE	25.7	0.9%	8.1%	15.3	0.6%	10.9%

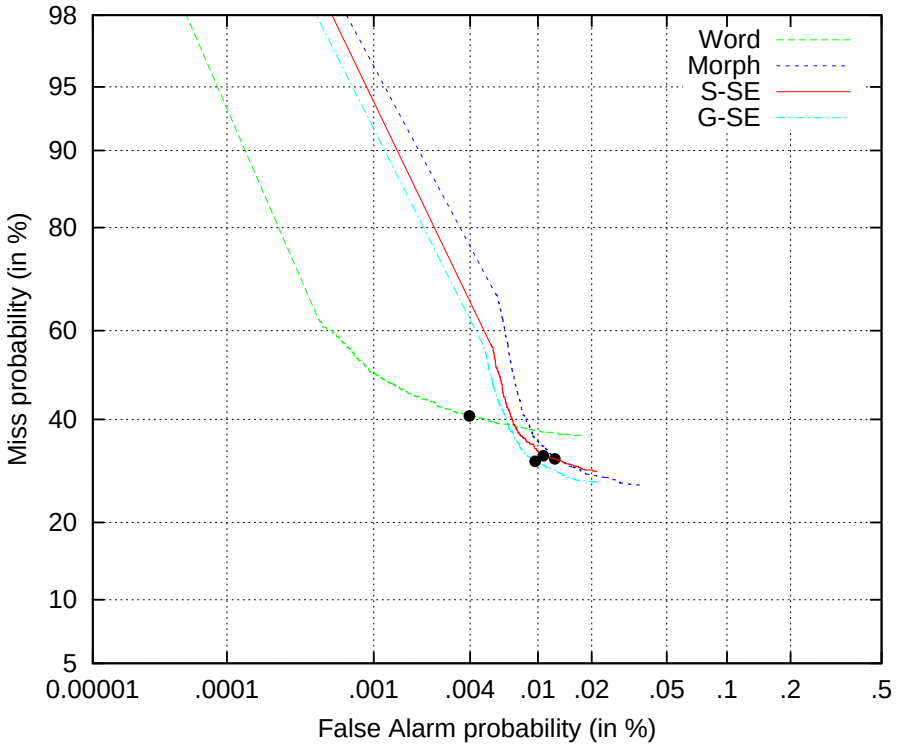

Figure 4.25. Comparison of word and various subword units on BN-2 corpus

can conclude that, as language modeling (and indexing) unit length gets smaller, recall rate increases (P_{miss} decreases), however precision gets lower (P_{FA} increases).

As presented in Table 4.16, use of subword units always improves the performance over words. Among the subword units best MTWV scores are obtained with the G-SE based index for both corpora. However, significance testing shows that words (200k),

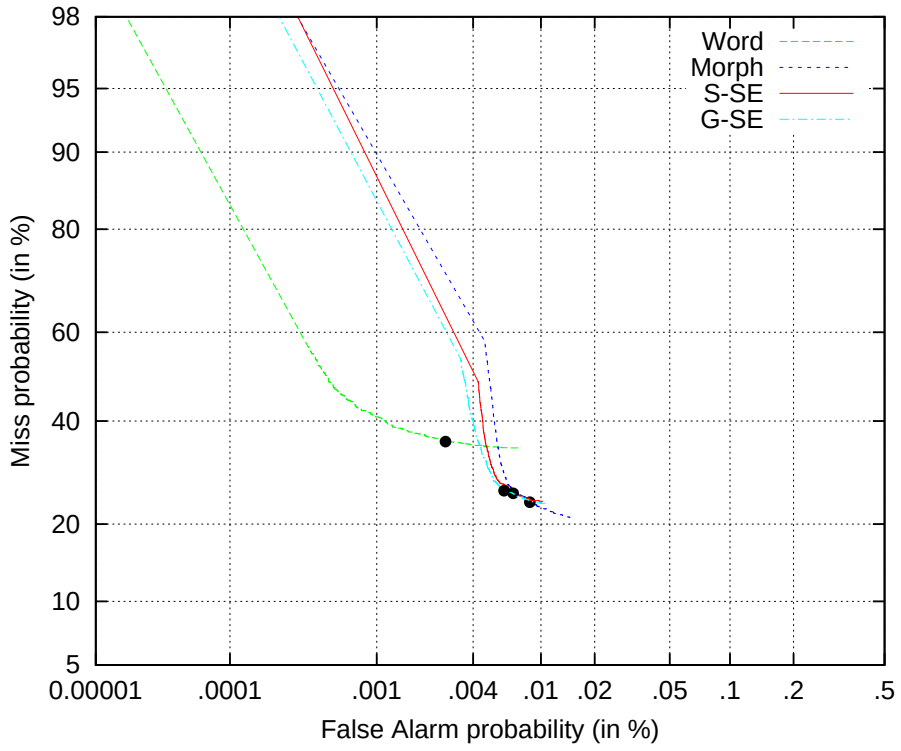

Figure 4.26. Comparison of word and various subword units on HI corpus

morphs and S-SEs do not have a difference for BN-2 corpus while G-SEs are significantly better than all of them. On the other hand, for HI corpus, all subword units are equal and better than words ($p < 0.05$).

Table 4.16. MTWV values of various units for both corpora. The values are computed over the whole query list (all), IV terms (IV) and OOV terms (OOV)

	BN-2			HI		
	all	IV	OOV	all	IV	OOV
Word (200k)	0.5525	0.7047	-	0.6173	0.8359	-
Morph	0.5592	0.6210	0.4425	0.6766	0.7740	0.4997
S-SE	0.5700	0.6470	0.3586	0.6780	0.7828	0.4248
G-SE	0.5920	0.6786	0.3737	0.6813	0.7788	0.4543

The DET curves are also computed for IV and OOV subsets of the query list as shown in Figure 4.27. For IV terms, longer units result in better performance. The

word model has the highest MTWV score for both corpora and a considerably superior DET curve. Since OOV words can not be located in the word based index, only OOV curves of subword models are plotted in the figure. The smallest length units, morphs, have the best performance in terms of MTWV on OOV queries.

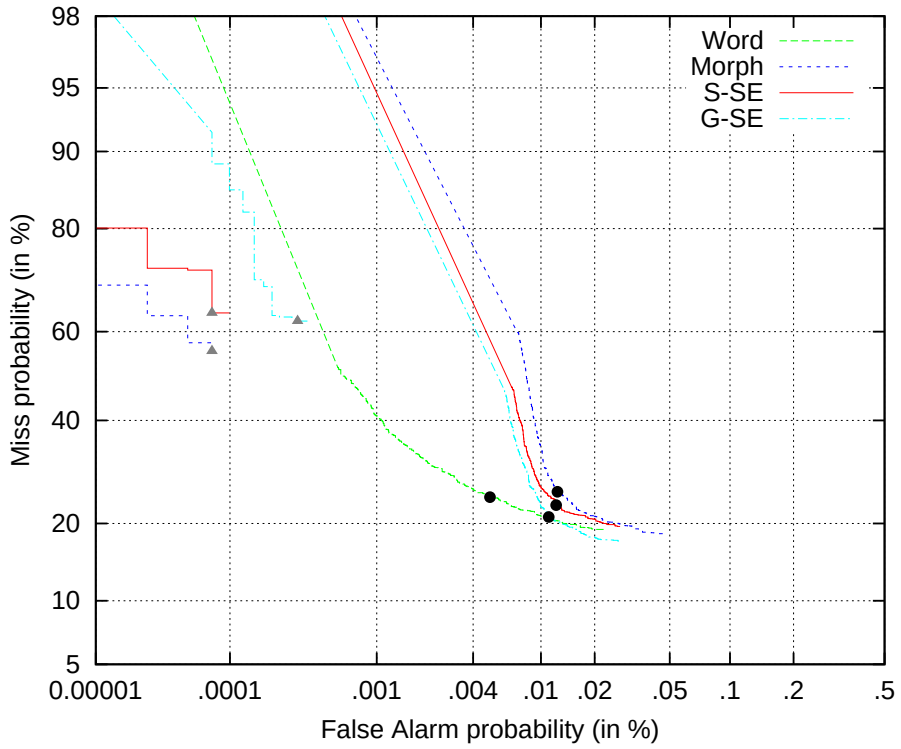

Figure 4.27. DET curves of various units on BN-2 corpus. The curves with a circle on their MTWV point are computed over IV terms. The curves with a triangle on their MTWV point are computed over OOV terms.

In overall, S-SE units have the highest MTWV values for both corpora. The reason can be explained as follows: Stem-endings are longer recognition units with low OOV rate. Thus, IV words have more emphasis on the result (as opposed to words). Since longer units perform better in IV set, stem endings are superior than morphs. Among stem-endings, G-SEs outperform S-SEs because their OOV rate on the query list is lower.

Lastly, we examine the cascade of word and subword indexes. Combining the results obtained above and cascading strategies explained in section 4.2.3, vocabulary

cascade corresponds to the combination of IV performance of word index and OOV performance of subword index. As justified above, this is the combination of advantages posed by two indexes. Search cascade results in a similar combination. Since word index is the best for IV words and morph index is the best for OOV words, their cascade provides the highest improvement. The DET curves of word, morph and their cascade are presented in Figure 4.28.

Paired t-tests also show the superiority of hybrids but do not report a significant difference between subword units in hybrids. For example, Table B.1 in Appendix B suggests that, search cascade of 200k vocabulary words with morphs, S-SEs or G-SEs are not significantly different at $p = 0.05$ for both corpora.

Significance testing results in Table B.1 also demonstrate that, for most of the hybrids (such as w-200+morph, w-200+S-SE, w-50+G-SE etc.) search cascading strategy yields significantly better scores than vocabulary cascading.

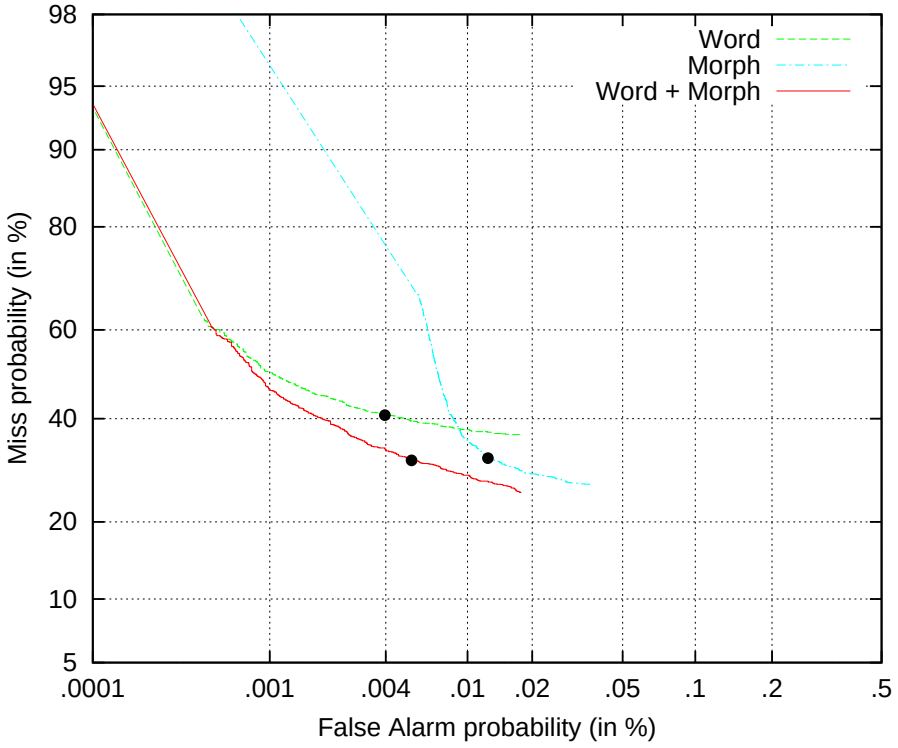

Figure 4.28. DET curves of word and morph indexes, along with their search cascade.

Tables 4.17 and 4.18 summarize all MTWV values and WER and OOV statistics of the second set of experiments. While G-SE has the highest score of individual indexes, search cascade of word and morph indexes demonstrates the best overall performance.

Table 4.17. A summary of MTWV values (w-50: Word index with 50k vocabulary, (V): Vocabulary cascade, (S): Search cascade)

	BN-2			HI		
	all	IV	OOV	all	IV	OOV
w-50	50.65	73.08	-	55.45	85.20	-
w-200	55.25	70.47	-	61.73	83.59	-
morph	55.92	62.10	44.25	67.66	77.40	49.97
w-50 + morph (V)	62.37	73.08	49.81	72.13	85.20	55.59
w-200 + morph (V)	62.71	70.47	44.25	72.43	83.59	49.97
w-50 + morph (S)	63.11	74.09	49.81	72.19	85.29	55.59
w-200 + morph (S)	**63.44**	71.42	44.25	**73.21**	84.56	49.97
S-SE	57.00	64.70	35.86	67.80	78.28	42.48
w-50 + S-SE (V)	61.13	73.08	40.94	71.05	85.20	48.76
w-200 + S-SE (V)	61.73	70.47	35.86	71.59	83.59	42.48
w-50 + S-SE (S)	61.94	74.25	40.94	71.32	85.62	48.76
w-200 + S-SE (S)	62.61	71.59	35.86	72.34	84.56	42.48
G-SE	**59.20**	67.86	37.37	**68.13**	77.88	45.43
w-50 + G-SE (V)	61.47	70.39	37.37	71.95	82.98	45.43
w-200 + G-SE (V)	61.86	70.47	37.37	72.40	83.59	45.43
w-50 + G-SE (S)	62.75	72.03	37.37	72.19	83.31	45.43
w-200 + G-SE (S)	63.07	72.01	37.37	72.93	84.30	45.43

Table 4.18. A summary of WER and OOV rates

	BN-2			HI		
	WER (%)	OOV-t(%)	OOV-q(%)	WER (%)	OOV-t (%)	OOV-q (%)
w-50	31.5	7.2	34.6	21.7	6.8	36.6
w-200	27.8	1.9	27.3	16.4	1.7	29.0
morph	26.1	0.6	7.6	13.8	0.5	10.2
S-SE	25.6	1.0	11.8	14.2	0.8	13.1
G-SE	25.7	0.9	8.1	15.3	0.6	10.9

4.5. An Application - SIGNIARY

Signiary (sign dictionary) is a Turkish sign language tutoring application where the user enters a word as text and retrieve videos of the related sign [49]. The word is searched within a collection of videos recorded from the Turkish Broadcast News for the Hearing Impaired, which is a subset of our HI corpus. The news video consists of three major information sources: sliding text, speech and sign. Figure 4.29 shows a snapshot of the program. The occurrences of a query are retrieved from the news videos, via speech and sliding text modalities. STD is used to exploit speech information. The retrieved videos are further analyzed to detect clusters among the signs that reflect pronunciation differences or sign homonyms.

Figure 4.29. An example frame from the news recordings.

In the news programme, the speaker signs as she speaks. However sign languages have their own grammars and word orderings; it is not necessary to have the same word ordering in a Turkish spoken sentence and in a Turkish sign sentence. Thus, the signing in these news videos should not be considered as Turkish sign language (Turk Isaret Dili, TID) but Signed Turkish.

The overall system works as follows: When user enters a query, the application returns several occurrences of the requested word via STD. If the resolution is high enough to analyze the lip movements, audio-visual analysis can be applied to increase accuracy. Then, sliding text information is exploited to control and correct the result of STD. The sign intervals are extracted by analyzing the correlation of the signs with the speech.

The STD setup in Signiary is similar to our first set of experimental setup. Acoustic and language model of ASR are the same as that of first set of experiments. Currently, subword based indexes are not employed in Signiary, only the baseline word-based LM works. Both lattice and one-best approaches are used as the output hypothesis. Evaluation is done over 15 videos from the HI corpus, which are selected to be concordant with the sign clustering part. The resulting precision-recall graph is shown in Figure 4.30.

Current system employs STD and sliding text recognition in cascade, which is implemented as follows: In STD part, the utterances whose relevance scores exceed a particular threshold are selected. In the sliding text part, STD hypotheses are checked with sliding text recognition. The intervals pointed by STD are scanned on the sliding text recognition output. In the interval, the word which is closest to the query (in terms of normalized minimum edit distance) is assigned as the corresponding text result. The normalized distance between search term and sliding text output is compared to another threshold. Those below the distance threshold are assumed to be correct. As presented in Figure 4.30, using both text and speech, the maximum attainable precision is 98.5%.

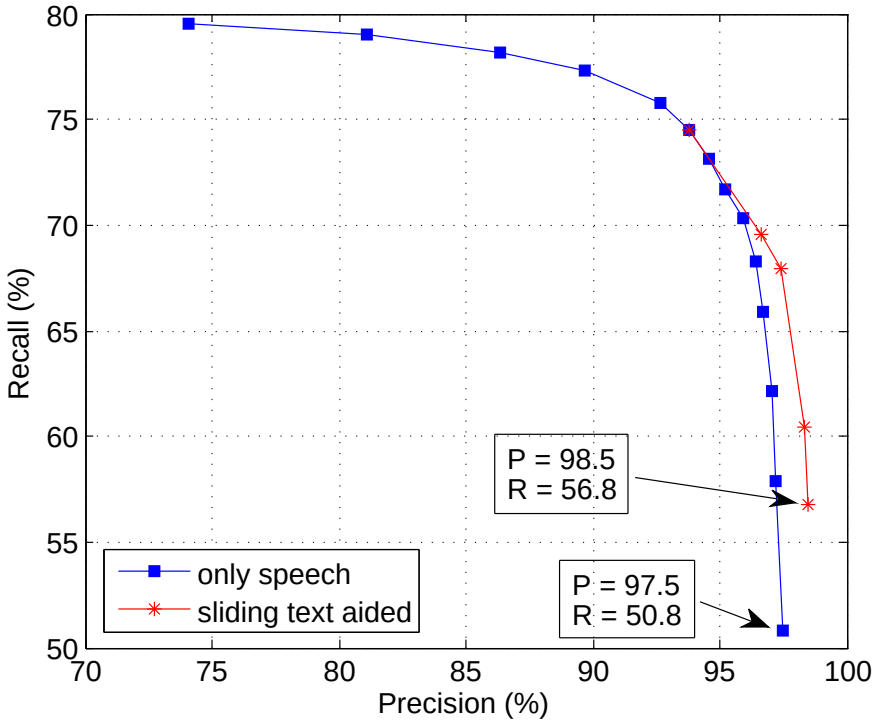

Figure 4.30. Precision-recall graphs, using only speech information and using both sliding text information.

5. SPOKEN DOCUMENT RETRIEVAL

Spoken Document Retrieval is the retrieval of spoken documents based on content. In our SDR task, we focus on indexing and retrieval of Turkish Broadcast News stories. In this chapter, we describe the baseline SDR system and and give the experimental results.

5.1. System Architecture

The block diagram of the SDR system is shown in Figure 5.1. Like STD, the SDR system incorporates three basic components: ASR, indexing and retrieval. ASR converts the spoken information to a textual information and it is the common component in STD and SDR. However, since SDR is based on content, instead of term matching, indexing and retrieval methods are different. As in the case of STD, the index is built offline whereas retrieval is performed after the query is submitted.

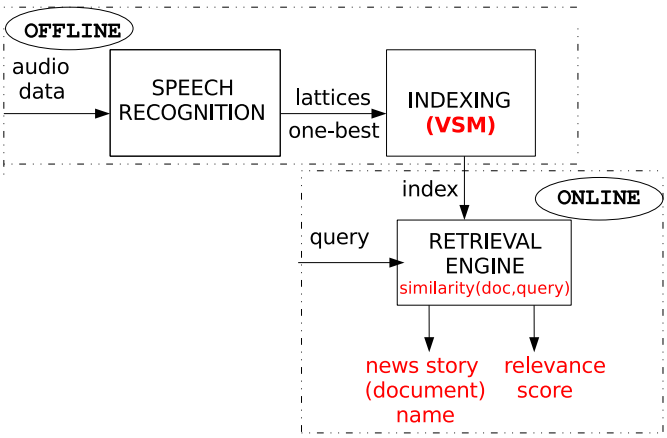

Figure 5.1. Block diagram of the SDR system

5.1.1. ASR

ASR prepares the spoken data for indexing by converting it into a textual representation. In SDR, we used the same HMM based LVCSR system as that of STD

(Section 4.1.1).

Like STD, subword based language modeling is useful for SDR, since it allows the OOV words to be recognized. In addition, subword based units are suitable indexing units to apply the stemming approach. Three different units are used for language modeling in SDR: words, G-SEs and S-SEs. Since the stem is usually enough to reflect the meaning for SDR, ending or suffixes are insignificant. Thus, smaller length counterparts of G-SEs and S-SEs, namely morphemes and morphs, are not used for language modeling in ASR.

We also investigate the effect of indexing alternative ASR hypotheses. Recall that, this is also a way of document expansion. Confusion networks are preferred for this purpose, since they are simple and we do not need to consider phrases as in the case of STD. In this approach, word posteriors are directly used as term frequencies. It may be useful to discard the words that have very low probability by setting a threshold. Thus if the sum of posteriors of a term is smaller than some threshold, it is assumed to be nonexisting in the document [42].

5.1.2. Indexing

After the automatic transcripts are obtained for each broadcast news video, they are segmented into news stories. News story segmentation is very useful in broadcast news retrieval since it creates homogenous smaller length documents. It is possible to find the desired information in a long text document by scanning, however this is not possible for audio; the whole file should be listened, which is so cumbersome. Thus, the spoken documents should not be so long and long ones should be segmented into pieces. While it is possible to define episodes based on time, segmentation by topic creates more meaningful segments, which is done by news story segmentation for broadcast news. We use the manually labeled news story boundaries for this purpose.

Automatic transcripts are indexed via Vector Space Modeling to retrieve news stories in SDR (Recall that, these transcripts are indexed to retrieve the occurrences

for STD). Each news story is represented with a vector of term weights. Term weights are calculated with the standard tf.idf approach, which can be expressed more formally as:

$$w_{ij} = tf_{ij} \times (1 + \log(\frac{N}{n_j})) \tag{5.1}$$

As explained in Section 2.2, stemming algorithms are commonly used in IR. We experiment with the following indexing units:

1. No stemming: All the words in the collection are indexed as they are. This is the baseline approach.
2. Fixed prefix: The first n characters of a word are used as the indexing unit ($n = 5$).
3. G-Stem+Ending: Words are segmented to stems and endings with a morphological parser as explained in Section 2.1.2.1. Both stems and endings are indexed.
4. G-Stem: Only the grammatical stems are used as the indexing units.
5. S-Stem+Ending: Words are segmented to stems and endings in an unsupervised manner with MDL algorithm as explained in Section 2.1.2.1. Both stems and endings are indexed.
6. S-Stem: Only the statistical stems are indexed.

In fact, using both stems and endings as the indexing unit contradicts with the traditional stemming approach. However, since the document frequencies of endings are very high, they are expected to have a negligible effect on the retrieval performance. In order to explore this, we index both stems and endings in addition to only stems.

Note that, stemming and document expansion (by including alternative hypotheses) are the only IR improving methods that are used in our baseline SDR system. Currently, we do not apply query expansion, dimensionality reduction, stopword removal etc.

5.1.3. Retrieval

For retrieval, the similarity is calculated between each news story and the query. News stories are ranked with descending similarity and returned to the user.

In VSM approach, after the query is submitted by the user, its vector is constructed. If the index is subword based (stemming approach is used), the query should be segmented into its subword units prior to the vector construction. Query term weights are computed with the tf.idf formula defined in Equation 5.1. Next, the cosine is calculated between the query vector and each of the document vectors. The smaller the angle, the similar the query and document are.

5.2. Experiments

5.2.1. Experimental Setup

5.2.1.1. Evaluation Metrics. Precision and Recall are the most commonly used metrics in IR evaluation. However, a single metric is more useful sometimes. Since we have a medium size collection, only a few documents may be related to the query (See Table 3.3). This makes Precision@10 and Precision@20 metrics unreliable. That's why the SDR performance is evaluated with "Mean Average Precision" and "Binary Preference" metrics in our experiments. MAP emphasizes returning the relevant documents earlier than the non-relevant ones. Definition and formula of MAP is given in Section 2.2.2.1. BPREF metric, which is defined in Section 2.2.2.2, is advantageous since it ignores the unjudged documents and prevents a possible bias on the measurement.

5.2.1.2. Training Corpora. The acoustic and language models of the ASR system are trained on the corpora of Second Set of STD experiments, explained in Section 4.4.1.1.

5.2.1.3. Test Corpus. The test set consists of 135 news programmes, which are segmented into 2425 news stories manually. Amount of the test data in terms of channel

and acoustic conditions is given in Table 5.1. SDR test corpus does not include hearing impaired news videos. All programmes include varying acoustic conditions, which makes the task more challenging.

Table 5.1. Amount of SDR test data (in hours) in terms of channel and acoustic conditions

Channel	f0	f1	f2	f3	f4	fx	Total
CNN	7.92	1.20	1.00	2.98	15.21	0.50	28.81
NTV	3.83	0.91	0.44	1.52	10.88	0.27	17.85
TRT2	0.43	0.13	0	0.62	1.06	0.02	2.26
VoA	15.93	0.42	3.16	2.71	2.93	0.07	25.22
Total	28.11	2.66	4.60	7.83	30.08	0.86	74.14

For all news stories, WERs are computed using word-based, G-SE based and S-SE based language models. Overall error rate of the test corpus in given in terms of Story Word Error Rate (SWER) and Programme Word Error Rate (PWER) as well as the WER. SWER is the average of WERs computed for each story and PWER is the average of WERs computed for each broadcast news programme. SWER is the most useful metric for SDR task because in SWER computation all stories have equal weights, independent of the length. Whereas in WER calculation, longer documents would have higher weights. As can be seen in Table 5.2, subword units result in a decrease of $\approx 2\%$ in all WER types.

Table 5.2. WER, PWER and SWER values for word and subword based language models.

	WER (%)	PWER (%)	SWER (%)
words	33.71	32.60	27.76
G-SEs	31.43	30.31	25.94
S-SEs	31.79	30.70	25.41

5.2.1.4. Topics & Queries. As mentioned in Chapter 3, a TREC-like topic set is created with short and terse topics, both of which are given in Appendix D. These short and terse topics are also used as queries.

Our human assessment system, which will be introduced in Section 5.2.1.5, allows users to submit their own queries about the given topic. We collect these to evaluate the system in case of real queries. This gives a more realistic perspective about user requests. In addition, we obtain a large number of queries by this way, which provides a more reliable statistical significance testing.

Some examples of user queries are shown below:

Topic 2:
Doğu ve Batı Almanya'nın karşılaştırılmasına ve birleşmesine ilişkin haberleri getir.
Queries:
berlin duvarı
berlin duvarı
doğu almanya ve batı almanya
dogu batı almanya berlin duvarı birleşme
doğu batı almanya berlin duvari sosyalizm
doğu batı almanya birleşme
doğu batı almanya birleşme
doğu batı almanya farklar birleşmesi
Topic 16:
Kalp hastalıklarının tedavisi ile ilgili yapılan çalışmalar elde edilen başarılar nelerdir?
Queries:
kalp hastalıkları kurtulma başarı öneri
kalp hastalıklarının tedavisi başarılar
kalp hastalıkları tedavisi çalışmalar başarılar
kalp hastaliklari ve tedavisi
kalp hastalık tedavi
kalp nakli

Table 5.3. SDR query analysis

	Short Topics	Terse Topics	User Queries
# of queries	27	27	232
# of tokens	220	104	774
# of types	164	95	310
tokens/query	8.15	3.85	3.33
# of OOV words	1	0	19
Tokenwise OOV rate	0.45%	0%	2.45%
Typewise OOV rate	0.61%	0%	5.48%

According to the statistics in Table 5.3, user queries include OOV words with a percentage of 2.45. However, when we examine the OOV words, we realize that most of them are typos. A common error is using English characters, instead of Turkish ones. Examples include *bogazici*, instead of *boğaziçi* and *hastaliklari*, instead of *hastalıkları*. Among the 19 OOV words, 17 words are in OOV set due to typos. From these observations we can conclude that, users tend to enter 3-4 words that are -probably- in recognition vocabulary as query terms.

5.2.1.5. Relevance Judgements. In addition to the initial topic labeling, news stories are re-judged by 11 human assessors, who are graduate students in Boğaziçi University and some other universities. Since three assessors' relevancy judgements are inappropriate they are discarded but their queries are used in the user query set. Assessors interact with the SDR system via a simple web page developed in javascript. The evaluation consists of three steps:

1. The login screen is displayed, which includes instructions and user login box. Users just enter their names, we do not use passwords. A screenshot of the login screen is shown in Figure 5.2.
2. After the user logs in, one of the short topics is displayed randomly. The assessor is asked to submit keywords to view the related news stories. Note that the topic is strict but queries are not. Namely, assessors are requested not to enter random queries or queries that are about any other topic but they are free to enter any

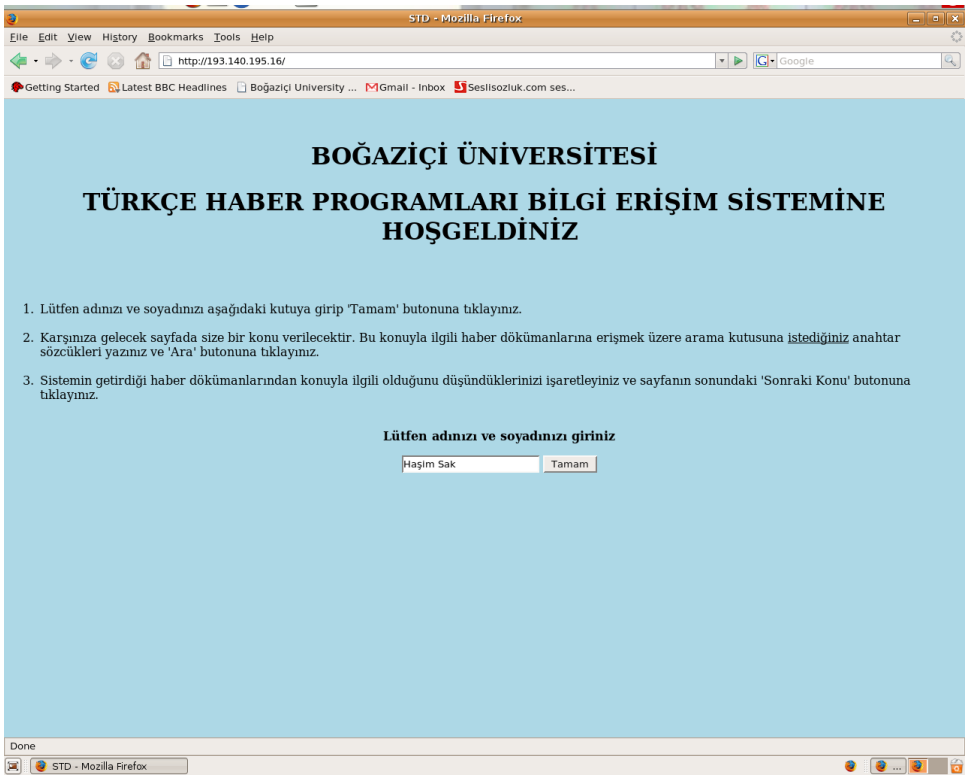

Figure 5.2. SDR human assessment system: login screen

words about the given topic.

3. After the query words are submitted to the system, they are searched in the index and results are returned to the user.
4. Each retrieved news story is associated with a tick box next to it. Assessors are asked to check the box if the news story is related to the topic. If not, the box is left unchecked. After all the stories are finished, the judgements are submitted to the server via clicking on the "Sonraki Konu" button at the end of the page, which also causes another topic to appear.

This process continues until all the topics are completed. However, assessors are not required to complete all the topics; they can quit after sending the judgements of a query.

We use the pooling concept in our human assessment system. Namely, the results are shown to the user from a pool at step 3 of the evaluation. Before the human assessment system is presented to assessors, we were able to evaluate the system using

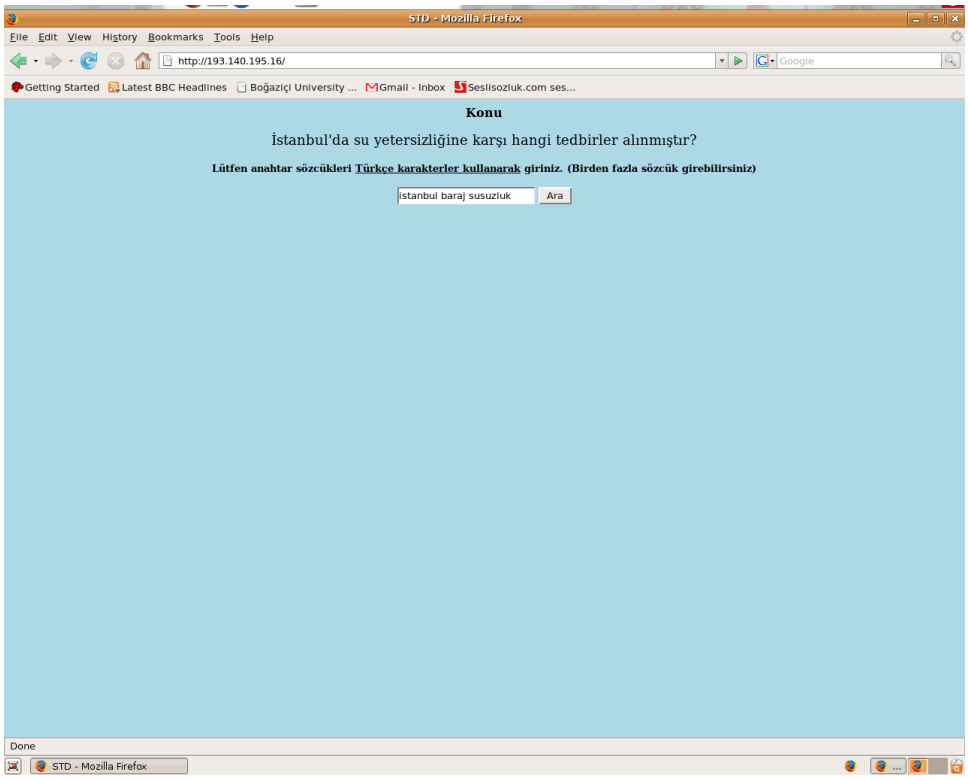

Figure 5.3. SDR human assessment system: one of the topics is presented to the assessor

topic labels (Recall that remember, each news story is labeled with a topic during topic segmentation). Fixed prefix method was the best performer in these preliminary experiments. That's why we base the pool construction on the fixed prefix method. The pools are constructed with the top 5 documents of five runs and initial topic labels:

1. Long queries are searched with the fixed prefix method in human transcripts.
2. Short queries are searched with the fixed prefix method in human transcripts.
3. Long queries are searched with the fixed prefix method in word-based recognition output.
4. Short queries are searched with the fixed prefix method in word-based recognition output.
5. The queries that are entered by the user are searched with fixed prefix method in human transcripts.
6. The news stories which are labeled as relevant to the query during manual segmentation are also added to the pool.

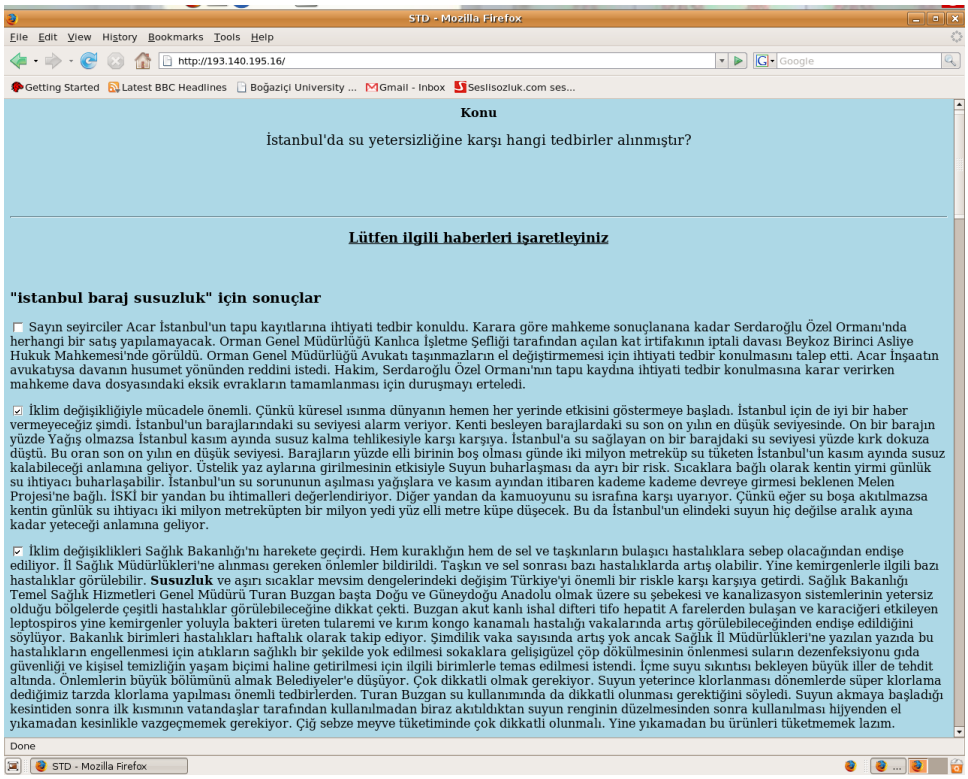

Figure 5.4. SDR human assessment system: relevant news stories are displayed to the assessor

5.2.2. Experimental Results

5.2.2.1. Effect of ASR and Stemming. To investigate the effect of ASR and stemming, we experiment with various language modeling units in ASR and various indexing units in indexation. Three different units are employed in ASR dimension: word based, G-SE based and S-SE based. We also index the manual transcripts and compare it with the ASR output. Different types of units that are used in indexing are: Fixed prefix, G-Stem+Ending, G-Stem, S-Stem+Ending and S-Stem. Along with the use of no stemming, we compare six stemming methods. For example, if the language model in ASR is G-SE based and indexing units are words (i.e. no stemming), then stems and endings are joined together to obtain words prior to indexing. Grammatical recognition and statistical indexing units (or vice versa) are not combined since this requires merging G-SE based recognition output into words and respliting words into statistical stems and endings. The missing items in the following tables (Table 5.4, Table 5.5 and Table 5.6) correspond to these kind of combinations. The baseline index

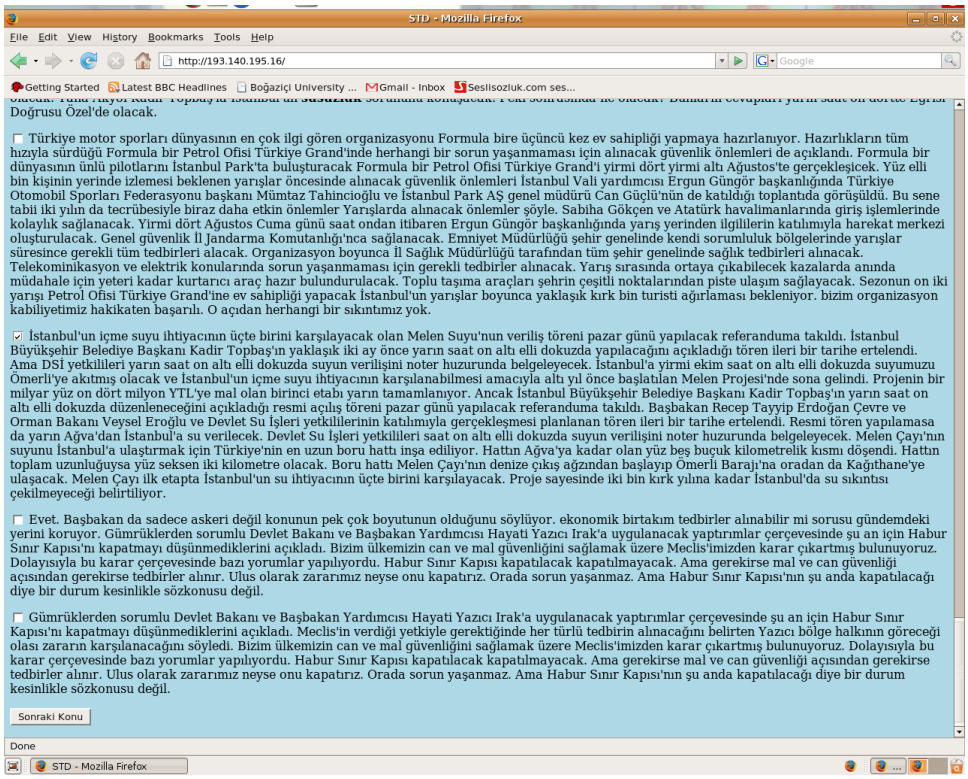

Figure 5.5. SDR human assessment system: assessor submits judgements and request another topic via clicking on the button

is constructed with word-based recognition and no stemming.

All of the experiments are run over three query sets: user queries, short topics as queries and terse topics as queries. Results are evaluated with significance testing. Student's t-test is used with a significance level of 5 %. Note that user queries provide a more realistic experimental setup. In addition, user query set includes 232 queries, which makes the significance tests more reliable. MAP and BPREF scores are presented in Tables 5.4, 5.5 and 5.6, respectively.

Using User Queries. As can be seen in Table 5.4, for user queries, (even though the absolute difference is minor) the reference transcripts are always superior to the ASR output and the stemming methods provide a remarkable gain over no stemming. While grammatical methods have slightly better MAP scores than the other subword based indexing units; statistical units perform better in BPREF.

Table 5.4. Scores of various transcriptions and indexing units over the user queries

MAP				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	30.66	28.64	29.03	28.52
Fixed-Prefix	36.98	35.19	35.06	34.83
G-Stem+Ending	38.08	36.33	35.97	-
G-Stem	38.33	36.14	35.99	-
S-Stem+Ending	37.75	36.15	-	36.01
S-Stem	37.55	35.98	-	35.08
BPREF				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	39.32	38.07	38.57	38.15
Fixed-Prefix	43.60	42.13	42.05	41.77
G-Stem+Ending	44.28	42.42	42.48	-
G-Stem	44.20	42.30	42.40	-
S-Stem+Ending	44.43	43.67	-	43.39
S-Stem	44.12	43.06	-	43.36

Significance testing results for user queries show that:

- All subword based indexing units provide gain over no stemming.
- Grammatical units are significantly superior to statistical units for particular cases (eg. Reference text) in MAP measure.
- Statistical units perform better than the fixed prefix for particular cases (eg. Word Recognition) in MAP measure.
- Statistical stemming is significantly the best performer in BPREF.
- Reference transcriptions work better than all kinds of ASR output.
- No significant difference is noticed between word based recognition, G-SE based recognition and S-SE based recognition.

Table 5.5. Scores of various transcriptions and indexing units over terse queries

MAP				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	36.31	33.89	35.38	33.49
Fixed-Prefix	45.56	44.50	44.81	44.06
G-Stem+Ending	46.93	45.41	44.55	-
G-Stem	46.85	45.31	45.63	-
S-Stem+Ending	44.48	42.44	-	42.03
S-Stem	43.46	42.05	-	42.26
BPREF				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	41.64	39.36	40.36	40.26
Fixed-Prefix	45.75	44.84	45.37	44.76
G-Stem+Ending	43.14	42.52	42.23	-
G-Stem	43.45	42.74	42.44	-
S-Stem+Ending	47.65	46.13	-	46.53
S-Stem	46.48	44.79	-	45.91

Using Terse Queries. Table 5.5 indicates that reference transcripts are always better for terse queries. Word based recognition, G-SE based recognition and S-SE based recognition have almost the same retrieval performance.

Among the indexing units, grammatical units achieve the highest MAP scores (except G-SE based recognition) and statistical units demonstrate the worst performance. Fixed prefix method lies between the two.

Significance testing results for terse queries show that:

- The recognition methods and reference transcripts are not significantly different
- For all recognition types, subword based indexing units are not superior to each other. Comparing nostemming approach to the stemming approaches, we noticed that in MAP nostemming is equal to the subwords except G-SE; G-SE is better

than nostemming.

- No stemming and stemming approaches do not have a significant difference in BPREF.

Table 5.6. Scores of various transcriptions and indexing units over short queries

MAP				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	37.97	35.35	35.73	35.55
Fixed-Prefix	45.99	43.61	43.47	44.38
G-Stem+Ending	45.70	44.57	44.54	-
G-Stem	46.39	45.21	42.81	-
S-Stem+Ending	45.58	44.87	-	45.05
S-Stem	45.23	44.87	-	44.94
BPREF				
	Reference	Word Recog.	G-SE Recog.	S-SE Recog.
No Stemming	41.64	39.54	41.20	40.96
Fixed-Prefix	44.30	42.94	44.75	45.06
G-Stem+Ending	42.80	40.77	40.94	-
G-Stem	42.66	41.29	40.98	-
S-Stem+Ending	47.56	46.44	-	46.72
S-Stem	45.57	45.58	-	45.87

Using Short Queries. As shown in Table 5.6, reference transcripts are always better for the short query set. Different language modeling units in ASR (word based, G-SE based and S-SE based) demonstrate equal retrieval performance.

Among the indexing units, subword based indexing is considerably superior than the no stemming approach. Although grammatical units seem to work better for most of the cases, there is no remarkable difference.

Significance testing results for short queries show that:

- As in the case of terse queries, the recognition methods and reference transcripts are not significantly different
- For all recognition types, subword based indexing units are not superior to each other. Comparing nostemming approach to the stemming approaches, we noticed that nostemming is equal to the subwords except G-SE; G-SE is better than nostemming for particular cases (not all cases, unlike terse queries).

Significance tests comparing the performance of short and terse queries show no difference. We do not compare them with the user queries, since they do not have equal number of samples. The reason of the lower MAP scores of user queries is probably the heterogeneity of queries. Namely, the user query set includes different number of queries for each query.

From these results we conclude that, regardless of the method all stemming approaches are extremely useful for SDR. On the other hand, decreasing the WER with subword based language models does not contribute to retrieval performance. The reason might be the robustness of SDR to OOV words (compared to STD). Recall that in SDR, it may be possible to retrieve the related documents even if the query includes OOV words.

5.2.2.2. Using Confusion Networks. Indexing the alternative ASR hypotheses is shown to be very helpful for STD. In this experiment, we investigate the effect of such method on SDR. Confusion networks are selected as the hypothesis expansion method because of their simplicity. The results are compared to the reference text and word based recognition output.

Tables 5.7, 5.8 and 5.9 present the results in MAP and BPREF for user queries, terse queries and short queries respectively. For all query types, CNs provide gain over one-best if no stemming is applied. No considerable improvement is noticed if any of the stemming approaches is applied. However, statistical tests show that use of CNs do not result in an increase in both MAP and BPREF scores for any case. The

reason can be explained as follows: SDR is not directly affected by an improvement in recognition. Namely, use of expanded hypotheses may provide the true hypotheses to be indexed but the overall SDR performance is not affected by this. From the document expansion point of view, CNs include phonetically similar hypotheses. On the other hand, document expansion with semantically related words may be more useful since SDR task is content based.

Table 5.7. Retrieval scores of reference text, one-best and CN indexation on user queries

MAP			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	30.66	28.64	29.00
Fixed-Prefix	36.98	35.19	34.52
G-Stem+Ending	38.08	36.33	35.08
G-Stem	38.33	36.14	35.07
S-Stem+Ending	37.75	36.15	36.22
S-Stem	37.55	35.98	35.78
BPREF			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	39.32	38.07	38.46
Fixed-Prefix	43.60	42.13	42.32
G-Stem+Ending	44.28	42.42	42.40
G-Stem	44.20	42.30	42.44
S-Stem+Ending	44.43	43.67	43.46
S-Stem	44.12	43.06	43.07

Table 5.8. Retrieval scores of reference text, one-best and CN indexation on terse queries

MAP			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	36.31	33.89	36.96
Fixed-Prefix	45.56	44.50	44.29
G-Stem+Ending	46.93	45.41	45.26
G-Stem	46.85	45.31	45.12
S-Stem+Ending	44.48	42.44	42.71
S-Stem	43.46	42.05	42.21
BPREF			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	41.64	39.36	40.37
Fixed-Prefix	45.75	44.84	45.56
G-Stem+Ending	43.14	42.52	42.11
G-Stem	43.45	42.74	42.25
S-Stem+Ending	47.65	46.13	46.78
S-Stem	46.48	44.79	45.22

Table 5.9. Retrieval scores of reference text, one-best and CN indexation on short queries

MAP			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	37.97	35.35	39.57
Fixed-Prefix	45.99	43.61	45.19
G-Stem+Ending	45.70	44.57	45.86
G-Stem	46.39	45.21	45.23
S-Stem+Ending	45.58	44.87	45.45
S-Stem	45.23	44.87	45.34
BPREF			
	Reference	Word Recog.	Word-CN Recog.
No Stemming	41.64	39.54	42.25
Fixed-Prefix	44.30	42.94	45.68
G-Stem+Ending	42.80	40.77	43.20
G-Stem	42.66	41.29	42.11
S-Stem+Ending	47.56	46.44	46.24
S-Stem	45.57	45.58	45.55

6. CONCLUSIONS

So far, spoken information retrieval has been investigated for several languages. However the problem was not explored for Turkish. This thesis is important in being the first speech retrieval study in Turkish. The experiments are performed on our Turkish Broadcast News Corpus.

We developed two types of systems for the retrieval of Turkish Broadcast News: a Spoken Term Detection system and a Spoken Document Retrieval system. They both combine ASR and IR components to retrieve spoken data. Although they employ the same ASR system to obtain textual information, the indexing and retrieval approaches are completely different. Since STD requires exact matching of a query, we used WFSA indexation to retrieve the occurrences. Whereas in SDR, the topic of the document has the major importance. For this reason the SDR index is built with VSM to facilitate the retrieval of related documents.

The classical approach of indexing the ASR output with text based IR methods is inadequate especially for agglutinative languages like Turkish. Various methods were attempted to alleviate the the effect of ASR errors on retrieval. Use of grammatical and statistical subword units as well as lattices introduced a significant gain to STD. The best scores of STD was obtained with the word-morph hybrid, which performed even better with term-specific thresholding in detection. In addition, use of alternative hypotheses offered a flexibility of selecting an operating point. We also introduced a method to change the operating point when using term-specific thresholds. For the second set of STD experiments, NIST-based tools are used for evaluation.

For the SDR task, we construct a baseline SDR collection. The test set is segmented into news stories, a TREC-like topic set is created and relevance judgements are made by human assessors. Experiments showed that, indexing the subword units provided gain over no stemming. Comparing the subword units, we did not observe a significant difference. Use of confusion networks, improved the system performance for

only a few cases. Unlike STD, expanding the best ASR hypothesis did not contribute to retrieval performance.

Comparing the two tasks, it can be concluded that, the methods to improve the retrieval performance resulted in different effects on STD and SDR. Since STD is based on term matching, its performance is directly affected by OOV words and WER variation. For this reason, subword based recognition and lattice based indexing were useful for STD. However it is not the case in SDR: none of them resulted in any contributions. On the other hand all of the stemming approaches (even a very simple one: pruning to a fixed length) provided improvements in SDR scores. The reason is the clustering effect of stemming: The words with a common stem are usually semantically related.

Future work includes the employment of lexical subword units in STD. Use of lexical stems as the indexing unit is expected to improve also the SDR performance since lexical stems are able to handle letter transformations in Turkish. In addition, disambiguating the morphological parser's output may be helpful to obtain the correct grammatical subword units and improve the results. Current SDR system utilizes the manually labeled story boundaries. It may be improved to automate the topic segmentation.

APPENDIX A: FINITE STATE MACHINES

Finite State Machines (FSM) are computing devices commonly used in language processing, specifically in speech recognition, morphological and syntactic parsing and machine translation.

An FSM (also called as a finite state automaton-FSA) is represented as a directed graph consisting of a set of nodes and arcs between the nodes. Below, a simple FSM is depicted [50].

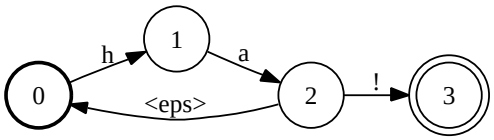

Figure A.1. A simple FSM

This FSM recognizes the letter sequence 'ha', repeated any times, ending with an exclamation mark such as "ha!", "haha!", "hahaha!" and so on. A recognizer FSM is also called an acceptor. Any other string is rejected by the FSM. The acceptor consists of 4 states. State 0 is the initial state and state 3, represented with a double circle, is the final state. An FSM can include more than one final states.

More formally, an FSM is a 5-tuple:

- Q: a finite set of N states $q_0, q_1, ..., q_N$
- $\sum$: a finite input alphabet of symbols
- q_0: The start state
- F: The set of final states
- $\delta(q, i)$: The transition matrix between states $(Q \times \sum : Q)$.

In a *deterministic* automaton, in each state, there is only one transition, given an input symbol. Namely, the algorithm has no choice, given the input, it knows what to do.

In a *finite state transducer* (FST), the transition table consists of symbol pairs, instead of single symbols. The transducer maps one set of symbols (inputs) to another set (outputs). Relating to finite state acceptor, the transducer maps the accepted string to another one. A simple transducer is illustrated in Figure A.2. If the input to this transducer is "haha!", then the output will be "xyzxy.".

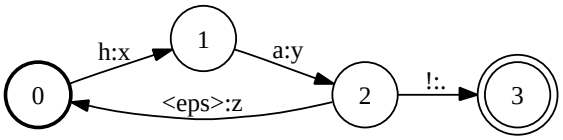

Figure A.2. A simple FST

An FST is a 5-tuple:

- Q: a finite set of N states $q_0, q_1, ..., q_N$
- $\sum$: a finite input-output symbol pairs
- q_0: The start state
- F: The set of final states
- $\delta(q, i)$: The transition matrix between states $(Q \times \sum : Q)$.

Some FSM operations and optimization algorithms are summarized below:

- Epsilon-Removal: Returns an FSM equivalent to the input FSM that is epsilon removed.
- Determinization: Creation of an equivalent deterministic FSM
- Minimization: Creation of an equivalent minimal (with minimal number of states and transitions) deterministic FSM
- Union: Creation of an FSM that has a choice of transitions from a node to the unioned FSMs.
- Composition: If the first FST transduces x to y and the second one transduces y to z, then their composition will transduce x to z.

A.1. Weighted Finite State Automata and Semirings

An FSM or FST is said to be *weighted* if the transitions are defined with probabilities (or costs) in addition to labels. WFSAs are commonly used in speech recognition since the weights handle the uncertainity of ASR hypotheses. The weighted version of the FST in Figure A.2 is shown below:

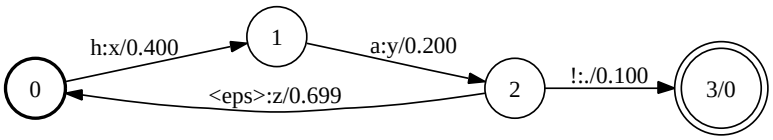

Figure A.3. A simple WFST

A system $(K, \oplus, \otimes, \bar{0}, \bar{1})$ is said to be a semiring if: $(K, \oplus, \bar{0})$ is a commutative monoid with identity element $\bar{0}$; $(K, \otimes, \bar{1})$ is a monoid with identity element $\bar{1}$; $\otimes$ distributes over $\oplus$ and $\bar{0}$ is an annihilator for $\otimes$ (for all $a \in K$, $a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$). Table A.1 lists some familiar semirings.

Table A.1. A list of familiar semirings ($a \oplus_{log} b = -log(e^{-x} + e^{-y})$)

	Set (K)	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Boolean	$0, 1$	$\vee$	$\wedge$	0	1
Probability	R_+	$+$	$\times$	0	1
Log	$R \cup \{+\infty, -\infty\}$	$\oplus_{log}$	$+$	$+\infty$	0
Tropical	$R \cup \{+\infty, -\infty\}$	min	$+$	$+\infty$	0

Two semirings often used in speech processing are the log semiring and tropical semiring. In both of them, the costs represent negative log probabilities. The tropical semiring selects the minimum cost weight as the best weight. Probability (or real) semiring has the probabilities as the arc weights. The best path is the one with the highest weight [45].

APPENDIX B: Significance Testing Results 1

Table B.1. Significance testing of STD experiments - Second Set (BN Corpus)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
w-50	1	.	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<
w-200	2		.	=	<	<	<	<	=	<	<	<	<	<	<	<	<	<
morph	3			.	<	<	<	<	=	<	<	<	<	<	<	<	<	<
w-50 + morph (V)	4				.	=	<	<	>	>	=	=	=	>	=	=	=	=
w-200 + morph (V)	5					.	=	<	>	>	>	=	=	>	>	=	=	=
w-50 + morph (S)	6						.	=	>	>	>	>	=	>	>	=	=	=
w-200 + morph (S)	7							.	>	>	>	>	=	>	>	>	=	=
S-SE	8								.	<	<	<	<	<	<	<	<	<
w-50 + S-SE (V)	9									.	=	<	<	=	=	=	<	<
w-200 + S-SE (V)	10										.	=	<	>	=	=	=	<
w-50 + S-SE (S)	11											.	=	>	=	=	=	=
w-200 + S-SE (S)	12												.	>	=	=	=	=
G-SE	13													.	<	<	<	<
w-50 + G-SE (V)	14														.	=	<	<
w-200 + G-SE (V)	15															.	<	<
w-50 + G-SE (S)	16																.	=
w-200 + G-SE (S)	17																	.

In the table, the operator (*) at location (a,b) shows the relation between a and b as "$a * b$". For example "$<$" at (3,13) states that morphs perform significantly worse than G-SEs. (w-50: Word index with 50k vocabulary, S-SE: Statistical Stem-Ending, G-SE: Grammatical Stem-Ending, (V): Vocabulary cascade, (S): Search cascade)

APPENDIX C: Significance Testing Results 2

Table C.1. Significance testing of STD experiments - Second Set (HI Corpus)

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
w-50	1	.	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<
w-200	2		.	<	<	<	<	<	<	<	<	<	<	<	<	<	<	<
morph	3			.	<	<	<	<	=	<	<	<	<	=	<	<	<	<
w-50 + morph (V)	4				.	=	=	<	>	>	=	=	=	>	=	=	=	=
w-200 + morph (V)	5					.	=	<	>	>	>	=	=	>	=	=	=	=
w-50 + morph (S)	6						.	<	>	>	=	=	=	>	=	=	=	=
w-200 + morph (S)	7							.	>	>	>	>	=	>	=	=	=	=
S-SE	8								.	<	<	<	<	=	<	<	<	<
w-50 + S-SE (V)	9									.	=	<	<	>	=	=	=	<
w-200 + S-SE (V)	10										.	=	<	>	=	=	=	<
w-50 + S-SE (S)	11											.	<	>	=	=	=	<
w-200 + S-SE (S)	12												.	>	=	=	=	=
G-SE	13													.	<	<	<	<
w-50 + G-SE (V)	14														.	=	<	<
w-200 + G-SE (V)	15															.	<	<
w-50 + G-SE (S)	16																.	<
w-200 + G-SE (S)	17																	.

In the table, the operator (*) at location (a,b) shows the relation between a and b as "$a * b$". For example "$<$" at (3,13) states that morphs perform significantly worse than G-SEs. (w-50: Word index with 50k vocabulary, S-SE: Statistical Stem-Ending, G-SE: Grammatical Stem-Ending, (V): Vocabulary cascade, (S): Search cascade)

APPENDIX D: Topics for SDR

D.1. Short Topics

1. Amerika'da Ermeni tasarısının görüşülmesine ilişkin haberleri getir.
2. Doğu ve Batı Almanya'nın karşılaştırılmasına ve birleşmesine ilişkin haberleri getir.
3. Dünyada hangi ülkelerde suikast olayları yaşanmış?
4. NATO ve NATO üyesi ülkelere ilişkin haberleri bul.
5. Ölüm cezasının uygulanmadığı ülkeler hangileridir?
6. Üniversitelerde meydana gelen intihar ve cinayet olaylarının sebepleri neler olabilir?
7. Son zamanlarda Türkiye'de enflasyon rakamları nasıl değişmiştir?
8. Döviz kurundaki önemli değişimleri ve zamanları bul.
9. Türkiye'de açlık sınırı ve sınırda yaşayan insanlar hakkında bilgiler getir.
10. Yaz aylarında aşırı sıcaklara karşı alınabilecek önlemler nelerdir?
11. En az on ölü ile sonuçlanan doğal afetleri getir.
12. Muhtemel İstanbul depremi ile ilgili yapılan çalışmalar ve alınan önlemler nelerdir?
13. İstanbul'da su yetersizliğine karşı hangi tedbirler alınmıştır?
14. Küresel ısınmanın etkilerinden örnekler bul.
15. Türk sineması ile ilgili çıkmış haberleri bul.
16. Türkiye'de işsizlik oranı ve eğitim düzeyi arasındaki ilişki nasıldır?
17. Türkiye'de ve dünyada son zamanlarda gerçekleşmiş uçak kaçırma hava korsanlığı vakalarını bul.
18. Kalp hastalıklarının tedavisi ile ilgili yapılan çalışmalar elde edilen başarılar nelerdir?
19. Türkiye'de kuş gribi vakaları hangi illerde görülmüştür?
20. Bilim adamlarınca kanseri önlediği düşünülen gıdalar hangileridir?
21. Abdullah Gül'ün Türk Kürt çatışması ile ilgili yaptığı açıklamaları bul.
22. Yirmi iki Temmuz seçimi öncesi Ak Parti'ye karşı düzenlenmiş yürüyüşler hangi

illerde yapılmıştır?

23. Tayyip Erdoğan'ın Amerika'ya ziyaretleri ve görüştüğü kişiler hakkında bilgiler getir.
24. Türkiye'nin Avrupa Birliği'ne adaylık süreci nasıl işlemektedir?
25. Türkiye'de ordu siyaset ilişkileri ne durumdadır?
26. Doping olaylarında adı geçen sporcular kimlerdir?
27. Türkiye basketbol ligi karşılaşmalarını getir.

D.2. Terse Topics

1. Amerika'da Ermeni tasarısı
2. Doğu Batı Almanya birleşmesi
3. Dünyada suikast olayları
4. NATO ve NATO üyesi ülkeler
5. Ölüm cezasının uygulanmadığı ülkeler
6. Üniversitelerde intihar cinayet
7. Türkiye'de enflasyon
8. Döviz kurunda değişimler
9. Türkiye'de açlık sınırı
10. Yaz sıcakları önlem
11. En az on ölü doğal afet
12. İstanbul depremi önlemleri
13. İstanbul'da susuzluk tedbirleri
14. Küresel ısınmanın etkileri
15. Türk sineması
16. Türkiye'de işsizlik eğitim ilişkisi
17. Uçak kaçırma hava korsanlığı
18. Kalp hastalıkları
19. Türkiye'de kuş gribi
20. Kanser önleyici gıdalar
21. Türk Kürt Abdullah Gül
22. Seçim Ak Parti yürüyüş

23. Tayyip Erdoğan'ın Amerika ziyaretleri
24. Türkiye Avrupa Birliği'ne adaylığı
25. Türkiye'de ordu siyaset ilişkileri
26. Doping olayları
27. Türkiye basketbol karşılaşmaları

REFERENCES

1. Jurafsky, D. and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition" *Prentice Hall*, 2000.

2. Chou, W. and B. H. Juang,"Pattern Recognition in Speech and Language Processing", *CRC Press*, 2003.

3. Huang, X. , A. Acero, H. W. Hon, "Spoken Language Processing: A Guide to Theory, Algorithm and System Development", *Prentice Hall*, 2001

4. Arisoy, E. , H. Sak, M. Saraclar, "Language Modeling for Automatic Turkish Broadcast News Transcription", in *Proc. Interspeech*, 2007

5. Aksungurlu, T. , S. Parlak, H. Sak, M. Saraclar, "Comparison of Language Modeling Approaches for Turkish Broadcast News",in *SIU, IEEE 16th Signal Processing and Communication Applications Conference*, 2008.

6. Whittaker, E. D. , J. M. V. Thong and P. J. Moreno, "Vocabulary Independent Speech Recognition Using Particles", in *Proc. ASRU*, 2001.

7. Carki, K. , P. Geutner and T. Schultz, "Turkish LVCSR: Towards Better Speech Recognition For Agglutinative Languages", in *Proc. ICASSP*, 2000.

8. Mengusoglu, E. and O. Deroo,"Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language", in *Proc. ICASSP*, 2001.

9. Dutagaci, H. , L. M. Arslan, "A Comparison of Four Language Models for Large Vocabulary Turkish Speech Recognition", in *Proc. Interspeech*, 2002.

10. Hacioglu, K. , B. Pellom, T. Ciloglu, O. Ozturk, M. Kurimo, M. Creutz, "On Lexicon Creation for Turkish LVCSR" in *Proc. Interspeech*, 2003.

11. Kurimo, M. , A. Puurula, E. Arisoy, V. Siivola, T. Hirsimaki, J. Pylkkonen, T. Alumae, M. Saraclar, "Unlimited Vocabulary Speech Recognition for Agglutinative Languages", in *Proc. HLT-NAACL*, 2006.

12. Arisoy, E. , M. Saraclar, "Lattice Extension and Rescoring Based Approaches for LVCSR of Turkish", in *Proc. Interspeech*, 2006.

13. Yates, R. B. and B. R. Neto, "Modern Information Retrieval", *ACM Press*, 1999.

14. Ponte, J. M. and B. Croft, "A Language Modeling Approach to Information Retrieval", in *Proc. SIGIR*, 1998.

15. Can, F. , S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, "Information Retrieval on Turkish Texts", in *Journal of the American Society for Information Science and Technology*, pp. 407-421, vol. 59, issue 3, 2008

16. Molgaard, L. L. , K. V. Jorgensen and L. K. Hansen, "CASTSEARCH - Context Based Spoken Document Retrieval", in Proc. *ICASSP*, 2007.

17. Manning, C. D. and H. Schutze, "Foundations of Statistical Natural Language Processing", *The MIT Press*, 2002.

18. TREC, "Text Retrieval Conference", *http://trec.nist.gov/*

19. Bitirim, Y. , Y. Tonta and H. Sever, "Information Retrieval Effectiveness of Turkish Search Engines", *Lecture Notes in Computer Science*, 2457, 93103.

20. Demirci, R. G. , V. Kismir and Y. Bitirim, "An Evaluation of Popular Search Engines on Finding Turkish Documents", in *Proc. ICIW, International Conference on Internet and Web Applications and Services*, 2007.

21. Ekmekcioglu, F. C. and P. Willett, "Effectiveness of Stemming for Turkish Text Retrieval", in*Program*, Vol. 34 No.2, pp.195-200, 2000.

22. Pembe, F. C. , "A Linguistically Motivated Information Retrieval System for Turkish", *MS Thesis, Boğaziçi University*, 2002.

23. Buckley, C. and E. M. Voorhees, "Retrieval Evaluation with Incomplete Information", in *Proc. of the 27th International Conference on Research and Development in Information Retrieval (ACM SIGIR)*, pp.25-32, 2004

24. NIST, "The Spoken Term Detection (STD) 2006 Evaluation Plan", *http://www.nist.gov/speech/tests/std*, 2006

25. Koumpis, K. and S. Renals, "Content-Based Access to Spoken Audio", in *IEEE Signal Processing Magazine*, 2005.

26. Lee, L. and B. Chen, "Spoken Document Understanding and Organization", *IEEE Signal Processing Magazine*, 2005.

27. Makhoul, J. , F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz, A. Srivastava, "Speech and Language Technologies for Audio Indexing and Retrieval", *Proc. of the IEEE*, 1999.

28. Thong, J. M. V. , P. J. Moreno, B. Logan, B. Fidler, K. Maffey and M. Moores, "SPEECHBOT: An Experimental Speech-Based Search Engine for Multimedia Content on the Web", in *IEEE Transactions on Multimedia*, 2002.

29. Hansen, J. H. L. , R. Huang, B. Zhou, M. Seadle, J. R. Deller, A. R. Gurijala, M. Kurimo and P. Angkititrakul, "SpeechFind: Advances in Spoken Document Retrieval for a National Gallery of the Spoken Word,", in *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp.712-730, 2005.

30. Vergyri, D. , I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark and W. Wang, "The SRI/OGI 2006 Spoken Term Detection System" in *Proc. Interspeech*, 2007, pp. 2393-2396.

31. Miller, D. R. H. , M. Kleber, C. Kao, O. Kimball, T. Colthurst, S. A. Lowe,

R. M. Schwartz and H. Gish, "Rapid and Accurate Spoken Term Detection" in *Proc. Interspeech*, 2007.

32. Logan, B. , P. Moreno and O. Deshmukh, "Word and subword Indexing Approaches for Reducing the Effects of OOV Queries on Spoken Audio", in *Proc. HLT*, 2002.

33. Logan, B. , J. M. V. Thong and P. J. Moreno, "Approaches to Reduce the Effects of OOV Queries on Indexed Spoken Audio", *IEEE Transactions on Multimedia*, Vol. 7, No. 5, October 2005.

34. Saraclar, M. and R. Sproat, "Lattice-Based Search for Spoken Utterance Retrieval", in *Proc. HLT-NAACL*, 2004

35. Mamou, J. , B. Ramabhadran, O. Siohan, "Vocabulary Independent Spoken Term Detection", in *Proc. SIGIR*, 2007.

36. Parlak, S. and M. Saraclar, "Spoken Term Detection for Turkish Broadcast News", in *Proc. ICASSP*, 2008

37. Hori, T. , I. L. Hetherington, T. J. Hazen and J. R. Glass, "Open-Vocabulary Spoken Utterance Retrieval Using Confusion Networks", *Proc. ICASSP*, 2007.

38. Matthews, B. , U. Chaudhari and B. Ramabhadran, "Fast Audio Search Using Vector Space Modelling", in *Proc. ASRU*, 2007.

39. Garofolo, J. , G. Auzanne and E. Voorhees, "The TREC Spoken Document Retrieval Track: A Success Story", in *Proc. Content Based Multimedia Information Access Conference*, 2000.

40. Kurimo, M. , V. Turunen and I. Ekman, "An Evaluation of a Spoken Document Retrieval Baseline System in Finnish",in *Proc. Interspeech*, 2004.

41. Kurimo, M. and V. Turunen, "To Recover from Speech Recognition Errors in Spoken Document Retrieval", in *Proc. Interspeech*, 2005.

42. Turunen, V. T. and M. Kurimo, "Indexing Confusion Networks for Morph-based Spoken Document Retrieval", in *Proc. SIGIR*, 2007.

43. Young, S. , D. Ollason, V. Valtchev and P. Woodland, "The HTK book (for HTK version 3.2)", *http://htk.eng.cam.ac.uk/*, 2002.

44. Stolcke, A. , "SRILM - An Extensible Lnaguage Modeling Toolkit", in *Proc. Interspeech*, pp.901-904, 2002.

45. Allauzen, C. , M. Mohri, M. Saraclar, "General-Indexation of Weighted Automata-Application to Spoken Utterance Retrieval", in *Proc. HLT-NAACL*, 2004.

46. Oflazer, K. , "Two-level Description of Turkish Morphology", in *Literary and Linguistic Computing*, vol. 9, pp. 137-148, 1994.

47. Gungor, T. , H. Sak, M. Saraclar, "Turkish Language Resources: Morphological Parser, Morphological Disambiguator and Web Corpus", *Language Resources and Evaluation*, 2008

48. Martin, A. , T. K. G. Doddington, M. Ordowski and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance", in *Proceedings of EuroSpeech*, 1997.

49. Aran, O. , I. Ari, E. Dikici, S. Parlak, P. Campr, M. Hruz, L. Akarun, M. Saraclar, "Speech and Sliding Text Aided Sign Retrieval from Hearing Impaired Sign News Videos", in *Journal on Multimodal User Interfaces*, 2008

50. Coleman, J. , "Introducing Speech and Language Processing", *Cambridge University Press*, 2005.