

Turkish LVCSR: Database Preparation and Language Modeling for an Agglutinative Language

Erhan Mengusoglu, Olivier Deroo

Faculté Polytechnique de Mons, TCTS Lab, Mons, Belgium

{mengus,deroo}@tcts.fpms.ac.be

Abstract

Turkish language is an agglutinative language. It is possible to produce a very high number of words from the same root with suffixes [1]. Language modeling for agglutinative languages needs to be different than modeling of languages like English. Such languages also have inflections but not as many as an agglutinative language. Techniques which can be used for modeling agglutinative languages are presented in this work.

Turkish is one of the least studied language for speech recognition. For this reason the first step for Turkish speech recognition is preparing a database. The texts to record the database were selected from television programs and newspaper articles. Selection criterion was to cover various subject and to create a phonetically balanced corpus. Additionally it is important to include as many different word as possible. The Speech Training and Recognition Unified Tool (STRUT)¹ has been used for training and testing systems for preliminary recognition experiments.

1. Introduction

Speech recognition has been one of the most studied fields in the past decades. Since current speech recognition systems for Large Vocabulary Continuous Speech Recognition (LVCSR) tasks are not very accurate, there is an increasing interest and research to improve the accuracy of the systems.

Researches on speech recognition are very different from each other. Some researchers are interested in acoustic analysis while others are interested in language modeling. The ultimate objective is to convert speech acoustic signal to the corresponding word sequence.

The best results for automatic speech recognition are obtained by applying statistical modeling techniques. Statistical modeling can be used for both acoustic analysis and language modeling in speech recognition. For modeling the speech acoustic signal it is common to use Hidden Markov Models (HMM) [8].

A typical HMM-based speech recognition system uses typically HMMs for each phoneme or for each

word. The main problem is to train these models for better matching between speech text and acoustic signal. If we have a sequence of acoustic vectors X for some acoustic data, the speech recognition problem consists in finding the best model M which maximizes $P(M|X)$ (Probability that the model M created the sequence of acoustic vectors X). The aim of training is to maximize this probability for correct matches between the model M and acoustic data X . Since HMM is unable to give directly this probability (it gives $P(X|M)$, probability of a model M created the acoustic data X), by using Bayes' law, it can be decomposed as [9]:

$$P(M|X) = \frac{P(X|M) \cdot P(M)}{P(X)}$$

where $P(X|M)$ is the likelihood of acoustic data X given model M , $P(M)$ is the prior probability of the model which can be obtained by a Language Model (LM) and $P(X)$ is the prior probability of the acoustic data which can be calculated from training data.

To create a language model we need a large text including ideally all possible word sequences in the language. A classical n-gram language model tries to find the probability of a word sequence of length n for the language [10]. For agglutinative languages it is very difficult to obtain a sufficiently large text because of the extremely large number of words. The language model will suffer from data sparseness and will not be reliable. So it is necessary to find a method more suitable than classical n-gram language modeling.

The paper is organized as follows: section 2 describes the Turkish morphology. Section 3 describes the Turkish database we collected. Section 4 gives some results obtained using the Speech Training and Recognition Unified Toolkit (STRUT) on the Turkish database we prepared. Section 5 gives some conclusions and future works.

2. Turkish Morphology

Turkish has an agglutinative morphology with productive inflectional and derivational suffixations. It is a member

¹<http://tcts.fpms.ac.be/asr/strut.html>

of the Altaic family of languages. The number of words is very high because of productive suffixations. According to [1] the number of distinct words in a corpus of 10 million words is greater than 400,000. This work also mentioned the data sparseness problem for such a large corpus.

Since there are so many words in the language, if we use a large vocabulary continuous speech recognition system for Turkish language, it is unlikely that the lexicon for speech recognition contains all the words. Consequently, the number of Out of Vocabulary (OOV) words will be large. There are some words that may not be appear in a very large text because of rare usage. An example for such a word is [2]:

OSMAN LILAŞTIRAMAYABİLECEKLERİ-
MİZDENMİŞSİNİZCESİNE

and its production obtained by breaking it down into its root and morphemes:

OSMAN +LI+LAŞ+TIR+AMA+YABİL+ECEK+LER+İMİZ
+DEN+MİŞ+SİNİZ+CESİNE

The meaning of this word is “as if you were of those whom we might consider not converting into an Ottoman”. As can be seen it is possible to say a sentence by a single word. The suffixes have derivational and inflectional effects on the root words. For the example above, the derivational suffixes change the meaning of the word as follows:

OSMAN	:Name
OSMANLI	:(the region) in which there are Osmans, Ottoman
OSMANLILAŞMAK	: Being Ottoman. (-MAK for infinitive form of the verb)
OSMANLILAŞTIRMAK	: Converting into an Ottoman.
...	

For more information on Turkish morphology there is a study for the description of Turkish morphology by finite state approach [2]. In this work morphological rules of Turkish language are redefined with a finite state approach.

The phonemes in the Turkish language are equivalent to letters. That means for each of the letters:

Vowels : a e i o ö ü
Consonants : b c ç d f g ğ h j k l m n p r s ş t v y z

there is a phoneme.

Turkish morphology exhibits vowel harmony. Suffixation is subject to vowel harmony. The first vowel of the suffix depends on the last vowel of the stem. Stem is the word without the suffix. Stem is different than the root

word while stems may include suffixes. A stem ending with a back vowel (a, ı, o, u) takes a suffix starting with a back vowel, a stem ending with a front vowel (e, i, ö, ü) takes a suffix starting with a front vowel.

2.1. Morphology based language modeling

Language modeling for agglutinative languages need to be different than language modeling of languages like English. Such languages also have inflections but not as many as an agglutinative language. Using the part-of-speech tags [6] that can be assigned to the words for language modeling might be a solution. But it is not useful since it causes loss of information for intermediate derivations. Because these derivations can contain markers for syntactic relationship between words.

One approach for modeling agglutinative languages is proposed by [3]. In this work, the roots and the endings of words are considered as language model entries. The procedure is:

1. Identify all possible endings for a language by using a vocabulary.
2. Extract the endings from all dictionary words. This can be done either by using a dictionary in which endings and stems are already defined or by processing the text to find endings and stems.
3. Take a sufficiently large text and by using the method in step 2, generate a text composed of stems and endings separated by white spaces.
4. Construct the vocabulary to be used for LM from the text generated in step 3.
5. For each stem, calculate a set of probability for the endings.
6. Generate an n-gram language model for any combination of stems and endings.

For Turkish, it is more easy to determine the root words, so in step 1, the root words are identified using a dictionary of root words.

In step 2 the possible list of endings will be determined. With the help of the morphological analyzer [2] the possible endings and stems can be found for a dictionary.

In step 3 from a large text, the stems and endings can be determined and separated by white spaces. Then it is possible to create an n-gram language model from the text obtained.

For the reason of the vowel harmony some endings are considered as equivalent, for example: *-dim* in *geldim* (I came) and *-dum* in *buldum* (I found) have the same function (first person past time). In implementation these suffixes can be represented as *-dHm* in which H means harmonize with the last vowel [2].

Another method for language modeling of agglutinative languages is proposed by [1]. This method is based on morphological structure of the Turkish language. The method is different than a classical n-gram language model since it tries to estimate the probability of the word from its composition. It does not use the history to find the probability of the word.

The aim is to model the distribution of morphological parses given the words and to seek a variable T , the sequence of morphological parse of a word, that maximizes the probability of the morphological parse given the word, $P(T|W)$:

$$\begin{aligned} \operatorname{argmax}_T P(T|W) &= \operatorname{argmax}_T \frac{P(T)P(W|T)}{P(W)} \quad (1) \\ &= \operatorname{argmax}_T P(T)P(W|T) \quad (2) \end{aligned}$$

Since $P(W)$, the probability of the occurrence of the word, is accepted as constant and is ignored. T includes the root form of word and all morphosyntactic features to determine the word so the probability of a word given its morphological parses:

$$P(W|T) = 1$$

and the equation (2) can be written as:

$$\operatorname{argmax}_T P(T|W) = \operatorname{argmax}_T P(T) \quad (3)$$

Then as trigram tag model $P(T)$ can be defined as;

$$P(T) = \prod_{i=1}^n P(t_i|t_{i-2}, t_{i-1}) \quad (4)$$

where

$$P(t_1|t_{-1}, t_0) = P(t_1)$$

$$P(t_2|t_0, t_1) = P(t_2|t_1)$$

Each t is a morphological structure and can contain one root word and a number of inflectional groups (IG). An example for IGs:

sağlam+laş+tır+mak

sağlam+Adj[^] DB+Verb+Become[^] DB+Verb+

Caus+Pos[^] DB+Noun+Inf+A3sg+Pnon+Nom

to cause (something) to become strong /

to strengthen (something)

The inflectional groups in the example are named according to their function. [^]DB is the derivational boundary. The inflectional groups that will be used in language model could be:

1. sağlam

2. Adj

3. Verb+Become

4. Verb+Caus+Pos

5. Noun+Inf+A3sg+Pnon+Nom

+Become:become verb, +Caus: causative verb, +Pos: positive polarity, +Inf: infinitive form of the verb, A3sg: 3rd singular person agreement, +Pnon: no possessive agreement, +Nom: nominative case.

There are also some other simplifications to alleviate the data sparseness problem, like: root word depends only on the other root words. In this case there will be an n-gram language model for the occurrence of a sequence of a root word sequence. The language model probability of root word sequences can be used in conjunction with the probability of the occurrence of a word from a root word and IGs.

3. Database Preparations

Turkish is one of the least studied language for speech recognition. For this reason the first of our work consisted in collecting a Turkish database. The texts to record the database were selected from television programs and newspaper articles. Selection criterion was to cover as many different subject as possible and to create a phonetically balanced corpus.

Text Selection

For an efficient segmentation procedure, we needed two different types of data: isolated words and continuous speech. For isolated words the 100 most frequently used words are selected [4]. For continuous speech 215 sentences are selected from 50 different subjects (Table 1). The subjects are chosen from the television news of Turkish national television TRT and from some newspaper articles. The text is available from Bilkent University². To include as many words as possible and to obtain an easy to read text were the objectives when choosing the sentences and the subjects.

Recording

For recording 20 native speakers were selected, 10 male and 10 female. Speakers were asked to read the prepared texts. Normal recording time for one speaker was about 25 minutes, 22 minutes for continuous speech and 3 minutes for isolated words. If there is an error in reading, speakers were asked to read it again. The erroneous parts were removed from the database during the post-processing procedure.

²ftp://cs.bilkent.edu.tr/pub/Turklang/corpus

Number of	
utterances (continuous speech)	215
isolated words	100
words in continuous speech	2160
different words in continuous speech	1564
different words in all recording	1618
male speakers	10
female speakers	10

Table 1: Text Statistics.

Speech was recorded in a quiet room for each speaker. Recording materials were a portable Sony DAT-recorder TDC-8 and a close talking Sennheiser microphone MD-441-U. Speech was digitally recorded at 32kHz sampling rate in stereo quality. It was transferred to a computer with a Zefiro digital sound card.

Data Processing

The recorded speech was down-sampled to 16kHz with 16 bit resolution in mono quality. Finally there were 215 utterances and 100 isolated words saved as a separate .WAV file for each of 20 speakers.

The isolated words for 10 of the speakers (5 male, 5 female) were manually segmented and phonetically labeled. The phoneme labels are selected as the letters used in written Turkish texts:

a, b, c, C, d, e, f, g, G, h, I, i, j, k, l, m, n, o, O, p, r, s, S, t, u, U, v, y, z.

The letters which differ from the letters of English alphabet are changed to the most similar English letter but they are presented in uppercase. This is done to simplify the processing of texts. 8 new phonemes were added to label the phonemes with a stop:

b1, c1, C1, d1, g1, k1, p1, t1.

The total number of phonemes used for labeling was 38 including the silence. The original letters converted to uppercase letters are:

ç : C, ğ : G, ı : I, ö : O, ş : S, ü : U.

The manual segmentation of isolated words was then used for the automatic segmentation of continuous speech. The sound file processing program **snorri**³ was used for phonetic labeling of data. The labeling of remaining data was done automatically by using the Speech Training and Recognition Unified Tool (STRUT).

³<http://www.babeltech.com/winsno/winsno.html>

4. Training and Recognition Results

STRUT software is used for speech analysis, acoustic model training and speech recognition purposes. There are independent programs for each step of training and recognition processes. STRUT includes the programs for hybrid HMM/ANN (Artificial Neural Network) speech recognition [9], [5]. The experiences explained here are based on this technique. The ANN used is a Multi-layer Perceptron (MLP). MLP outputs are phoneme a posteriori probabilities for a given acoustic feature vector. Feature vectors are RASTA [14] feature vectors. Recognition is performed by Viterbi [15] decoding.

There are also some STRUT programs to make segmentation and phonetic labeling of speech data given some labeled data. The resulting labeled data can be used to label new data.

The recognition performance of the system is calculated by counting the correctly recognized words.

For the experiments, the Turkish database described in section III was used. The results are shown in **Table 2**. The first results are obtained by training a MLP with phonetically labeled isolated words from 10 speaker. There were 100 words for each speakers. Then the isolated words speech from 10 other speakers was used to test the speech recognition system. The results are not as good as expected, this is because of not using enough data to train the MLP. Recognition accuracy for continuous speech should be better after some more training and introducing a well defined language model.

	Word Error Rate
Training Data	0.6%
Test Data	6.6%

Table 2: Results for isolated word recognition.

5. Conclusions and Future Works

Since the languages are very different from each other it is important to use language specific properties in a speech recognition system. This properties can be used to select a training database, to create a language model which uses the specific morphological structure of the language.

In this paper we have presented two different language modeling techniques for agglutinative languages. They are promising to give better results for Turkish large vocabulary speech recognition. We have also described the collection of Turkish database.

We are also planning to use the language modeling techniques studied in this paper, to improve the efficiency of confidence measures [13] [12] defined on acoustic models of speech.

6. References

- [1] D. Hakkani-Tür, K. Oflazer, G. Tür, “Statistical Morphological Disambiguation for Agglutinative Languages”, Technical Report, Bilkent University, 2000.
- [2] K. Oflazer, “Two-level Description of Turkish Morphology”, *Literary and Linguistic Computing*, 9(2):137-148, 1994.
- [3] D. Kanevsky et. al., “Statistical Language Model for Inflected Languages”, US patent no: 5,835,888, 1998.
- [4] C. Yılmaz, “A Large Vocabulary Speech Recognition System for Turkish”, MS thesis, Bilkent University, 1999.
- [5] O. Deroo, “Modèles Dépendants du Contexte et Méthodes de Fusion de Données Appliqués à la Reconnaissance de la Parole par Modèles Hybrides HMM/MLP” PhD Thesis, Faculté Polytechnique de Mons, 1998.
- [6] J. Carlberger, V. Kann, “Implementing an Efficient Part-of-speech Tagger”, *Software Practice and Experience*, 29, 815-832, 1999.
- [7] G. Tür, D. Hakkani-Tür, K. Oflazer, “Statistical Modeling of Turkish for Automatic Topic Segmentation”, Technical Report, Bilkent University, 2000.
- [8] L. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, *Proceedings of IEEE*, 77(2):257-286, 1989.
- [9] H. Bourlard, N. Morgan, “Connectionist Speech Recognition: A Hybrid Approach”, Kluwer Academic Publishers, 1994.
- [10] A. Borthwick, “Survey Paper on Statistical Language Modeling”, Technical Report, Proteus project, New York University Computer Science Department, 1997.
- [11] B. Hoffman, “The Computational Analysis of the Syntax and Interpretation of Free Word Order in Turkish”, PhD Dissertation, University of Pennsylvania Institute for Research in Cognitive Science.
- [12] E. Mengusoglu, C. Ris, “Use of Acoustic Prior Information for Confidence Measure in ASR Applications”, to appear in *Proceedings of EUROSPEECH*, 2001.
- [13] G. Williams, “Knowing What You Don’t Know: Roles for Confidence Measures in Automatic Speech Recognition”, PhD Thesis, Department of Computer Science, University of Sheffield, 1999.
- [14] H. Hermansky, N. Morgan, “RASTA Processing of Speech”, *IEEE Transactions on Speech and Audio Processing*, 2(4), 578-589, 1994.
- [15] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”, *IEEE Transactions on Information Theory*, 13(2), 260-269, 1967.