A New Decoder Design For Large Vocabulary Turkish Speech Recognition

Onur Çilingir

Signal Proc. and Remote Sensing Group
TÜBİTAK BİLTEN, Ankara, TURKEY
onur.cilingir@bilten.metu.edu.tr

Mübeccel Demirekler

Dept. of Electrical & Electronics Eng.,
Middle East Technical University, Ankara, TURKEY
demirek@metu.edu.tr

Abstract

An important problem in large vocabulary speech recognition for agglutinative languages like Turkish is the high out of vocabulary (OOV) rate caused by extensive number of distinct words. Recognition systems using words as the basic lexical elements have difficulty in dealing with such virtually unlimited vocabulary. We propose a new time-synchronous lexical tree decoder design using morphemes as the lexical elements. A key feature of the proposed decoder is the dynamic generation of the lexical tree according to the morphological rules. The architecture emulates word generation in the language and therefore allows very large vocabularies through the defined set of morphemes and morphotactical rules.

1. Introduction

State of the art speech recognition systems yield satisfactory recognition accuracies for languages like English where vocabularies necessary for the required text coverage does not exceed a few hundred thousand words. There have been limited number of attempts for speech recognition for other languages which are morphologically more complex[1, 2]. Turkish is one of the least studied languages.

Besides the difficulties arising from lack of speech and text databases, previous studies also point out the problems with the vocabulary selection and out of vocabulary rates[3]. Turkish is an agglutinative language. Word formation plays a major role in sentence setup. New words can be generated by concatenating inflectional and derivational suffixes to roots/stems. The suffixes determine the meaning of the word and its function in the sentence. The morphotactics are so generative that total number of distinct words is very high. For practical considerations vocabulary requirement can be accepted as virtually infinite.

Conventional large vocabulary speech recognition systems recognize an utterance by searching among a fixed recognition dictionary. The motivation behind this study is to replace the fixed recognition dictionary with a dynamic structure which, during the decoding process, procedurally generates lexical entries according to a set of morphemes and related morphological rules.

In this study, a time-synchronous isolated word HMM decoder, which uses a dynamic lexical tree generation algorithm has been designed and implemented. This system is supplied with a set of morphemes consisting of roots and suffixes. Suffixation rules have been specified through encoded morphotactics definitions. The system is tested and compared to a fixed vocabulary tree decoder with equivalent word span.

This paper is organized as follows. In section 2 Turkish morphology is described from the speech recognition perspective. Section 3 introduces the proposed decoding algorithm and explains its details. The results of the experiments conducted on the implemented decoder design is given in section 4. Section 5 concludes the study and gives future research ideas.

2. Turkish Morphology

Turkish morphology contains two types of morphemes, the *roots* and the *suffixes*. Turkish words are generated by concatenation of suffix morphemes to roots. A suffixed root is called a *stem* and stems can take further suffixes. This architecture can be demonstrated by a frequently referred example[4].

Osmanlılaştıramayabileceklerimizdenmişsinizcesine

which is a word obtained from concatenation of a root followed by 12 morphemes. This adverb can be translated into English as: “as if you were of those whom we might consider not converting into Ottoman”.

Roots are categorized according to their part of speech (POS) like verb, noun, adjective, adverb, pronoun etc., which define their function in the sentence. A root can have more than one POS associated with it (homonyms). POS of a root can be changed by insertion of *derivational* suffixes whereas *inflectional* suffixes are responsible for case and context adjustments.

gel (verb: “come”)
gel + mAk = gelmek (→ noun: “to come”, derivation)
gel + dH = geldi (→ “he/she came”, inflection)

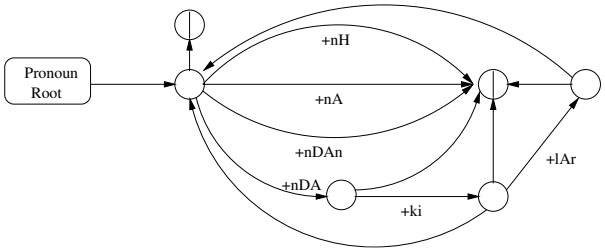

Figure 1: Suffix FSN for pronoun POS

The morphological base of this study is taken from the two-level analysis of Turkish morphology by Oflazer[4]. The two levels are the lexical and surface levels. Lexical level defines the structure and relation of the morphemes while surface level is for the orthographic realization. The morphotactics defined by the lexical level is described by use of *Finite State Networks(FSN)*.

Figure 1 shows the FSN used for the pronoun roots. It is a reduced and simplified version of the original FSN. The arcs denote the suffixes. Arcs without labels denote null transitions. Nodes with a vertical bar represent possible word ends. Note that the network contains a feedback which theoretically allows words of infinite length, which is impossible in practice. So a limiting mechanism is employed in the decoder while traversing the FSNs.

The FSN defines the suffixation combinations for any root of the corresponding POS, but the suffixes are in symbolic form and actual orthographic word decomposition, which is necessary for acoustic model setup, is supplied by another module. Surface realization module, applies the surface level rules of the two-level morphology to any given morpheme compound and determines the orthographic output in Turkish letters. The surface level rules vary from general context dependent rules like vowel harmony to special cases like vowel/consonant drop under certain conditions. Following two examples demonstrate the concept.

- The vowel immediately preceding the continuous tense suffix "+Hyor" is deleted.
- Volatile consonants (y,s,n) following by consonant ended stems drop.

In order to incorporate word generation functionality into decoding process, FSNs are determined for four POS: verb, noun, adjective and pronoun. The pronoun FSN has been given in Figure 1 and pronoun and adjective FSN are derived from the broad type of *nominal*(noun) FSN. Table 1 shows the statistics of the FSN definitions. It should be noted that the FSNs are simplified and actual suffixing is much more complex than the implemented FSNs.

The columns in table 1 denote the number of nodes, arcs and total number of word combinations that can be generated from a root for the given FSN. However the network design is different than figure 1. Nodes correspond to morphemes (suffixes) and arcs determine transitions. Final column shows that for each noun root, 7133 inflectional-derivational distinct variant words can be generated. This number illustrates the productivity of Turkish morphology. For verbs number of combinations becomes 1412. Although verb FSN is more complex with higher node and arc numbers, noun FSN has feedback connections as described for pronoun. For networks with feedback, each node is restricted to be activated once during traversal.

Table 1: FSN file statistics for each POS

FSN Name	# Nodes	# Arcs	Total Combinations
Verb	44	100	1412
Noun	40	90	7133
Adjective	20	41	25
Pronoun	6	16	14

After determination of the morphotactics through the implemented FSNs, selection of roots has been made from a small Turkish corpus of 1 million words, taken from different sources like electronic news and literature. Each word had morphological parse(s) from which the most frequent 200 roots have been selected. The POS distribution for these roots are given in table 2.

Total number of POS for the 200 roots is 277, i.e. larger than number of roots, since some roots have homonyms and their meaning fall into different morphological categories. When this 200 word root set is substituted in the FSN in table 1, total number of distinct word span is found to be $726,353$, which leads to a set of $641,758$ unique words.

3. The Decoder

The idea behind the proposed decoder design is to change the lexical units of conventional decoders from words to morphemes. Therefore instead of a fixed word dictionary, a set of root words and morphological word generation information are used. Decoder design is based on a time-synchronous tree decoder with beam-pruning. Consequently it has been given the name *Morphological Dynamic Tree Decoder (MDTD)*.

Table 2: POS distribution for the 200 roots

POS	Count	POS	Count
Noun	87	Adjective	37
Pronoun	12	Adverb	28
Verb	74	Postposition	13
Determinative	6	Conjunction	13
Number	5	Interjection	2
Total			277

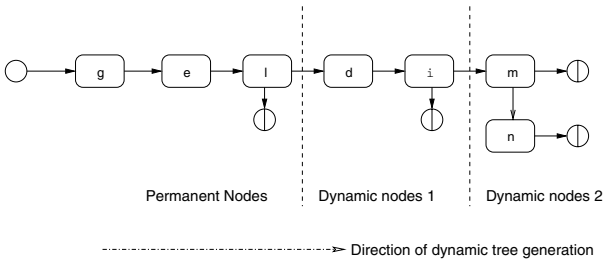

Figure 2: Dynamic tree generation

MDTD tree is made up of nodes. Each node corresponds to a lexical element, i.e. a phoneme. Nodes have both acoustic and morphological references. Acoustic reference points to an HMM whereas the optional morphological link determines the morphological decompositions of the partial word the node corresponds to.

Key concepts of MDTD are the pruning and procedural lexical tree generation. For very large vocabularies, a conventional time-synchronous tree decoder has a large lexical tree. However with pruning, in a single word recognition, only a very small fraction of its nodes are visited, the rest are temporarily redundant. MDTD starts the recognition of the utterance with a minimal tree and develops the tree for those hypotheses which are promising. Figure 2 illustrates the dynamic tree generation concept. For simplicity, nodes have been assigned monophone labels instead of triphones.

MDTD uses two trees, the lexical tree, which is illustrated in figure 2 and the morphological tree. Morphological tree is created from the FSNs for different POS. Each leaf node of the morphological tree corresponds to a valid suffixation of a POS root. During system setup morphological tree nodes are generated by enumerating the suffix combinations of the POS FSNs and adding the new entries to the appropriate nodes. The simple exemplary morphological tree for the node generation in figure 2 is shown in figure 3. Suffixation has been applied to the root verb “gel”.

In the initialization phase of MDTD, the set of root words and their POS tag(s) are read from a file. From the triphone decomposition of the root, necessary lexical nodes are generated and associated with their acoustic model reference. However, the root itself is not sufficient for the orthographic realization of a word. The following example shows a possible case:

kitap + A = kitaba

which demonstrates that the root word (and its pronunciation) is affected by the addition of the suffix “+A”. Since the network generation is assumed to be one-directional (which is the direction of the time-synchronous search), this feedback constitutes an important problem. A mechanism should be invented so that the “kitab” version of

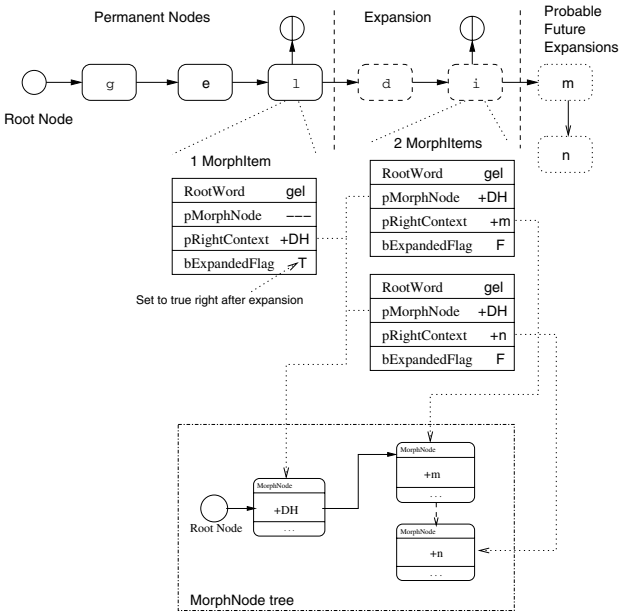

Figure 3: Dynamic network expansion example. Both lexical and morphological trees are shown

the root “kitap” is created in the lexical tree before it is actually needed by the added “+A”. The solution has been achieved by a lookahead technique which enumerates all right contexts of a ’stem+morpheme’ compound and adds all different versions to the tree. When the initialization of the tree is completed, lexical network is in its minimal form, the node structure in this form is recorded and the existing nodes are signed as *permanent* nodes.

After completion of the lexical tree setup, system becomes ready for recognition. Decoding process is based on Viterbi algorithm applied on a huge compound HMM. The main difference is seen when a terminal node is reached. For terminal nodes with morphological right contexts, *expansion* process takes place. New lexical nodes are created and inserted in the lexical tree, with appropriate acoustic model links and morphological tree references, according to the FSN and surface realization rules. The key point in this process is that only those lexical nodes which survive the pruning are given the opportunity to have expansion. Figure 3 shows the details of the expansion and dynamic generation processes. After each recognition session, lexical network is reset to its minimal state by deletion of non-permanent nodes added during recognition.

4. Experimental Results

In order to test the proposed decoding algorithm, an isolated word decoder based on MDTD method has been implemented. Training data consists of 70 minutes of single speaker speech containing 1000 sentences. For acoustic model training HTK toolkit has been used[5].

Table 3: Comparison of SLTD and MDTD decoders

	SLTD	MDTD
Total Recognition Time	615secs	202secs
Memory Requirement	420MBytes	10 MBytes
Recognition Accuracy	87.35%	87.35%

Mel-cepstrum based features are used for speech parameterization where each observation vector consists of 12 cepstrum coefficients, energy and their first and second derivatives. A total of 5767 logical and 867 physical triphone HMMs are achieved by application of *Decision Tree Clustering* as a refinement method using state tying.

Aim of the experiments is to test the proposed decoding approach which integrates Turkish morphology into recognition process. Since the key aspect of the proposed design is in the lexical representation domain of decoding and no statistical improvements on acoustic modeling and statistical language modeling are claimed, recognition accuracy is expected to be same as the baseline system. Experiments are based on evaluation of resource and computational requirements of MDTD algorithm with respect to the baseline system. Therefore, in addition to MDTD decoder, a fixed vocabulary time-synchronous tree decoder has been implemented and used as the baseline system, which is given the name Statical Lexical Tree Decoder(SLTD). Both decoders are built over the same core code so that the only difference between them is the dynamic lexical tree generation algorithm described in the previous sections.

With the implemented morphological rules and the selected set of 200 roots, a total number of 640K distinct words can be generated. Since both decoders are expected to yield identical recognition accuracy, a feasible subset of the total words is used as the sample test set. This test set consists of 1060 isolated words, selected randomly from the base set according to a selection probability determined by the word histogram in the text corpus. Each root is represented by a certain number of its suffixed variant words. (Verb:6, noun:6, adjective:2, pronoun:2, others:1). Table 3 shows the results of the experiments. It should be noted that SLTD is run on a fixed vocabulary containing 640K words (creating 1.7M nodes in the lexical tree) whereas MDTD has a set of 200 root words and morphological information and uses a dynamic vocabulary *virtually identical* to SLTD. Therefore SLTD requires 420MBytes of memory while MDTD has a peak usage of 10Mbytes.

Table 3 shows that total recognition process time of SLTD is much larger than MDTD. In fact, MDTD algorithm has the additional task of generating its lexical network at decoding time. This task is associated with morphological evaluations and memory allocation overheads. Therefore MDTD is expected to run slower than SLTD. The inverse relation in processing times can explained by the huge memory requirement of SLTD, which probably causes efficiency loss due to problems in memory paging and caching. It is possible to change recognition time and memory requirement of MDTD by adjusting the beam pruning threshold.

5. Conclusions

In order to achieve acceptable OOV rates, a word based fixed vocabulary system needs a very large word dictionary. However such big dictionaries yield impractical memory requirements. The proposed decoding algorithm, MDTD, alternatively uses a dynamic lexical tree, which is generated toward promising word hypotheses by using morphological information. With a minimal set of root words, very large number of words can be recognized. Experiments show that the implemented MDTD system can successfully recognize words from a vocabulary of 640K, generated from 200 root words, using modest memory. Another advantage of the design is that when a new root is added, all morphological derivatives of that root automatically become recognizable, which requires extra effort and resource with conventional recognizers.

Although isolated word recognition implementation has been chosen for simplicity, MDTD is applicable to continuous recognition as well. Another important contribution to the algorithm is supposed to be the addition of statistical language modeling. The morphological approach is well suited for LM application since it reduces the complexity of the statistics due to drastically reduced symbol set. Other research areas may be dynamic adaptation of lexical tree and handling special case morphological phenomena.

6. References

[1] O. Kwon, K. Hwang, J. Park, “Korean large vocabulary continuous speech recognition using pseudo-morpheme units”, Proc. EUROSPEECH, 483–486, 1999.

[2] C. Martins, J.P. Neto, L.B. Almeida. “Using partial morphological analysis in language modeling estimation for large vocabulary Portuguese speech recognition”, Proc. EUROSPEECH, 1603–1606, 1999.

[3] K. Çarkı, P. Getutner, T. Schultz, “Turkish LVCSR: Towards Better Speech Recognition for Agglutinative Languages”, ICASSP, 3:1563–1566, 2000.

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

[5] S. Young, “The HTK Hidden Markov Model Toolkit: Design and Philosophy”, Technical Report, Dept. of Engineering, Cambridge Univ., 1993.