On Lexicon Creation for Turkish LVCSR

Kadri Hacioglu, Bryan Pellom
University of Colorado at Boulder, USA

Tolga Ciloglu , Ozlem Ozturk
Middle East Technical University, Turkey

Mikko Kurimo , Mathias Creutz
Helsinki University of Technology, Finland

Abstract

In this paper, we address the lexicon design problem in Turkish large vocabulary speech recognition. Although we focus only on Turkish, the methods described here are general enough that they can be considered for other agglutinative languages like Finnish, Korean etc. In an agglutinative language, several words can be created from a single root word using a rich collection of morphological rules. So, a virtually infinite size lexicon is required to cover the language if words are used as the basic units. The standard approach to this problem is to discover a number of primitive units so that a large set of words can be created by compounding those units. Two broad classes of methods are available for splitting words into their sub-units; morphology-based and data-driven methods. Although the word splitting significantly reduces the out of vocabulary rate, it shrinks the context and increases acoustic confusibility. We have used two methods to address the latter. In one method, we use word counts to avoid splitting of high frequency lexical units, and in the other method, we recompound splits according to a probabilistic measure. We present experimental results that show the methods are very effective to lower the word error rate at the expense of lexicon size.

1. Introduction

Turkish is an agglutinative language. Several words can be created from a single root word using a rich collection of morphological rules. This is a major problem for large vocabulary continuous speech recognition (LVCSR), since one needs a vocabulary of virtually infinite size for a good coverage of the input language. Table 1 demonstrates the severity of the problem. The results are obtained over 792 sentences spoken by 20 speakers using the University of Colorado (CU) Sonic speech recognizer [1] ported to Turkish [2] with a vocabulary size limited to 30,000 words. There is a significant increase in the word error rate (WER) as the number of unknown words increases in the input. Substitutions for or deletions of out-of-vocabulary (OOV) words during recognition may also cause the in-vocabulary words to be misrecognized. To improve coverage, and hence the WER, we need to discover a number of primitive units so that a large set of words can be created by compounding those basic units. This can be achieved by splitting a large collection of words (in a principled way) into their parts and discovering a common set of primitive units that can be used to create all of the words. The splitting can be done at several levels; namely, phoneme, syllable, and morpheme levels. Only the morpheme level word splitting is suitable for LVCSR, since it is straightforward to determine the word boundaries. "Meaningful" units may also be useful in subsequent language processing stages for post-correction or even for better semantic parsing of imperfectly recognized sentences. The word boundary problem is illustrated in the following Turkish language example.

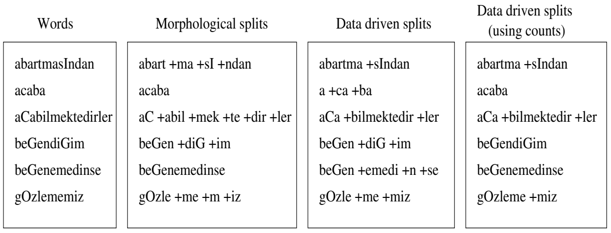
Table 1: Impact of out-of-vocabulary (OOV) words on speech recognition. Here, the WER is counted separately for sentences of variable amounts of OOV words.

# OOV words	# sentences/words	OOV%	WER
0	125/820	0.0%	34.6%
1	250/1672	15.0%	52.9%
2	244/1802	27.0%	65.0%
3	115/820	42.1%	78.4%
≥ 4	58/497	50.1%	90.1%

sentence:	kemanlar gIcIrdadI piyano CInladI
phonemes:	K E M AA NN LL A RH GG I C I RR D AA D I P IY Y AA NN O CH I NN LL A D I
syllables:	ke man lar gI cIr da dI pi ya no CIn la dI
morphemes:	keman +lar gIcIr +da +dI piyano CIn +la +dI.

Although our previous experiments have considered phoneme-based [2] and syllable-based [3] recognizers (the most recent phoneme error rate is 26.0% and syllable error rate is 43.6%) we do not know how to extend those systems to LVCSR. Therefore, in this paper, we present our work on splitting Turkish words into "morpheme-like" units. To do so, we use a Turkish morphological analyzer developed at the Middle East Technical University and a data driven word splitter developed at the Helsinki University of Technology [4].

Our Turkish morphological analyzer has been developed on PC-Kimmo, a free morphological parser based on Kimmo Koskenniemi's model of two-level morphology. We have used TurkLex, a set of PC-Kimmo rule specifications for Turkish morphology developed by Oflazer [5], together with some modifications for the morphological analysis operations. The lexical form has been used for different surface morphemes in the rule generation system. Such a morpheme usage results in ambiguities; a



Words	Morphological splits	Data driven splits	Data driven splits (using counts)
abartmasIndan	abart +ma +sI +ndan	abartma +sIndan	abartma +sIndan
acaba	acaba	a +ca +ba	acaba
aCabilmektedirler	aC +abil +mek +te +dir +ler	aCa +bilmektedir +ler	aCa +bilmektedir +ler
beGendiGim	beGen +diG +im	beGen +diG +im	beGendiGim
beGenemedinse	beGenemedinse	beGen +emedi +n +se	beGenemedinse
gOzlememiz	gOzle +me +m +iz	gOzle +me +miz	gOzleme +miz

Figure 1: Examples of word splits with different methods

post-processing algorithm has been developed to convert the lexical forms derived by PC-Kimmo into surface forms.

Our data-driven approach to the word-splitting problem is to find the optimal segmentation of a given source text into subword units according to a defined cost function. This cost function is based on the minimum description length (MDL) principle and it is the sum of coding the training text and the used codebook (vocabulary), and we used an iterative search algorithm [4] to minimize the cost. The cost of sub-word units is defined based on their likelihood estimates and the cost of codewords on the number of bits needed to code all its characters. Recently we have extended the algorithm to take into account the counts of the units to be split. In Figure 1, examples of some word splits using different splitting methods are exhibited.

The advantage of the data-driven approach is its straightforward portability to other languages and significant saving of expensive expert labor needed to create a morphological analyzer with high coverage. As illustrated in Table 2, the coverage (or recall) of the data driven approach is very high. However, its accuracy (or precision) is low. Experiments have shown that the Turkish morphological analyzer has slightly lower coverage but significantly better accuracy.

Table 2: Recall and precision rates. The rates are computed over a randomly picked set of 1500 words by comparing the word splits against linguistically correct morpheme boundaries. Recall is the ratio of the number of split words to the number of words to be split. Precision is the ratio of correct splits to the number of split words.

Method	Recall	Precision
Morphological	93.8%	98.2%
Data Driven	99.4%	32.8%

It is uncommon to use n-grams beyond bigrams or trigrams for language modeling in the state-of-the-art speech recognizers, due to computational and memory limitations. This means that the LM context in most practical systems covers at most two words. Splitting divides a word into several units. Therefore, the effective context to be used in LM modeling shrinks with the number of splits. This is expected to have a negative impact on the system's performance. To alleviate the problem we explore data-driven compounding after splitting. We provide experimental results that show about 5% relative improvement. We also present experimental results to show the impact of different word splitting methods on language modeling and recognition of Turkish.

Language modeling problems of Turkish have also been studied in [6-8]. However, to the best of our knowledge, there is only one work [9] that has reported results on the WER of Turkish LVCSR.

2. Morphological word splitting

Morphological word splitting is based on the morphological analysis of Turkish. PC-Kimmo[1], a freely available morphological analyzer based on Kimmo Koskenniemi's model of two-level morphology, has been used in doing the morphological analysis. The rules and the lexicon developed by Oflazer [5] have been integrated with PC-Kimmo and the lexicon has been enlarged to improve the coverage over the text used in the experiments.

The morphological analysis procedure follows the two-level phonology of Turkish as an agglutinative language. Each word in the language, namely the surface realization of the formation, has a corresponding lexical realization representing the structure of the word formation in terms of its morphological components, roots/stems, suffixes or prefixes (e.g. lexical: ev+lAr → surface : ev0ler → evler).

Morphological word splitting is achieved by first providing the word lists to PC-Kimmo. Based on the defined rules and the root and suffix/prefix lexicons, a list containing the corresponding lexical forms is generated. In accordance with the process of word formation in terms of the two-level morphology, in most of the cases, there is no one-to-one correspondence between the lexical forms and the words that appear in surface forms in the text. However, it is essential to extract the morphemes of the words as they exist in the text. Thus, at the second stage a post-processing of the lexical forms has been carried out.

The post-processing algorithm also refers to the word construction rules of Turkish. It has been developed as a separate software that takes the list produced by PC-Kimmo as its input. We follow almost a backward morphological analysis to obtain the surface counterparts of the morphemes. The post-processing yields the words split in their surface forms (e.g. evler → morph-analysis : ev+lAr → post-process : ev+ler). Post-processing algorithm compiles a multitude of phonetic rules of agglutination in Turkish to perform the inverse operation.

Currently there are approximately 35,000 roots and we still need to expand the list for more roots and some rare morphemes to further improve the coverage.

3. Data-driven word splitting

To automate the word splitting process and to optimize the subword unit selection for the given task, we applied an entirely unsupervised algorithm recently proposed in [4]. The algorithm operates language-independently and discovers the optimal set of subword strings based on any large text corpus or word list of the target language. The chosen units can be stems, prefixes, or suffixes, as long as they occur frequently enough in the corpus. The text-based search algorithm does not guarantee the optimality for speech recognition, but it is intuitively motivated for statistical language models to pick units with many training samples and for speech decoding to have long units to minimize confusions. The method has also been successfully applied for language modeling and large vocabulary speech recognition in Finnish[4,10].

We call the data-driven sub-word units morphs, because they

[1] http://www.sil.org/computing/catalog/pc-kimmo.html

resemble morphemes, the smallest meaning-bearing units of language. The algorithm learns the morphs from the corpus by recursively splitting each word until the defined MDL cost function is minimized [4]. The words are processed in random order and all the possible split locations are considered. The whole corpus is reiterated until the model converges to a minimum cost. The cost function is defined as the sum of the cost of coding the whole corpus and the cost of coding the used codebook. The cost of coding a morph (or sub-word unit) is defined using its probability in the corpus and the cost of a codeword by its length in characters.

The data-driven splitting has a clear advantage over the traditional morphemes, because it can easily be applied to any language without the need of manual work and expertise involved in construction of the morphological splitting rules. If the training corpus used to obtain the automatic morph set is large enough, the expected coverage on any new related test data and even on unseen word forms is high, too. Naturally, it is not guaranteed that the obtained morphs have one-to-one correspondence to the true morphemes of the language, but for many speech recognition applications this is not a problem. In some experiments [4], however, the resemblance to the true morphemes is high, and we expect that the obtained units can also be helpful in speech understanding.

4. Data-driven recompounding of lexical units

In this section we describe a data-driven method, which has been extensively studied in [11,12], for compounding lexical units. The word splitting methods mentioned earlier yield many short suffixes. Although the splitting is necessary for a significant reduction in OOV rates, it creates two important problems:

- shorter lexical units have higher error rates (as demonstrated in [12])
- the span of an n-gram LM with splits is significantly shorter (n is typically fixed to 2 or 3 regardless of the lexical units)

Due to the above problems, it is highly likely to lose a part of the performance gained from reducing the OOV rate. We believe that the loss can be partially recovered by compounding frequently co-occuring lexical units in a principled way and adding them to the lexicon as new lexical units. However, in turn, the compounding process creates a larger lexicon with increased acoustic confusibility that might incur some additional performance loss.

The measure that we used for compounding two lexical units w_i and w_j is the geometrical average of the direct and reverse bigrams:

$$\begin{aligned} M(w_i, w_j) &= \sqrt{P_f(w_j|w_i) P_r(w_i|w_j)} \\ &= \frac{P(w_i,w_j)}{\sqrt{P(w_i)P(w_j)}} \end{aligned} \quad (1)$$

The measure M is lower bounded by 0.0 and upperbounded by 1.0. The value of 1.0 means that the pair is a perfect candidate for compounding, since the probability of "w_j is preceded by w_i" and the probability of "w_i is followed by w_j" are both 1.0. Our implementation is slightly different from [11,12]. We first train a bigram LM for the initial lexicon. Then we compute M for the all bigrams that appeared in the LM. We select a subset of the bigrams for compounding by using a threshold on M and use that subset to create a new lexicon. Then we modify the training text, train a new LM and choose another subset of bigrams to compound. One can repeat the process a number of times to create longer units.

5. Experimental Results

In this section we first present language modeling and then speech recognition results. The text corpus we use has been collected from Turkish newspapers. It consists of approximately 2M words. The audio corpus was collected at METU. It was created from a set of phonetically balanced sentences. The text for the audio corpus was created as the Turkish translation of the first 2000 sentences of TIMIT database. Additional sentences are recorded to ensure that the most frequent 5000 tripones in Turkish were covered. It should be noted that the audio text is out-of-domain when compared to the news text. For a detailed description of text and audio corpora see [2].

Table 3 and Table 4 summarize in-domain and out-of-domain (OOD) LM results. In-domain test set is created by picking randomly 20% percent of the news text. The OOD test set is the text of audio data with which we carried out speech recognition experiments. It consists of 792 sentences. For in-domain LM experiments the LM is trained over the remaining 80% news text. For OOD experiments the whole news text is used to train the language models. All LMs are trigram LMs, and they are smoothed using the Witten-Bell method with no cut-offs.

Table 3: In-domain language model (LM) results. The LM quality is defined as the percent of test corpus trigrams that are present in LMs. The numbers in parentheses show the number of suffixes in the lexicon.

Method	Lexicon size	OOV rate	$PP*$	LM quality
No split	60,000	9.6%	380	33.1%
Morphological	44,526 (386)	2.4%	1,110	72.21%
Data Driven(DD)	15,722(4,147)	2.8%	890	57.4%
DD w/counts	48,828 (6,594)	3.9%	630	37.6%

Table 4: Out-of-domain language model (LM) results. The LM quality is defined as the percent of test corpus trigrams that are present in LMs. The numbers in parentheses show the number of suffixes in the lexicon.

Method	Lexicon size	OOV rate	$PP*$	LM quality
No split	60,000	15.2%	682	18.5%
Morphological	49,342 (389)	4.8%	1,927	58.3%
Data Driven(DD)	15,932 (4,288)	5.4%	2,012	42.0%
DD w/counts	49,429 (7,008)	7.3%	1,991	20.6%

Table 5: Recognition Results

Method	WER	WER*
No split, 30,000	61.1%	59.4%
No split, 60,000	56.3%	54.3%
Morphological	57.2%	55.4%
Data Driven	54.8%	52.2%
Data Driven, compound, itr=1	52.5%	50.0%
Data Driven, compound, itr=2	52.0%	49.6%
Data Driven, w/counts	46.6%	43.0%

* with unsupervised speaker adaptation

Although the word splitting significantly reduces the OOV rate, which is a major reason for high recognition error rates, the OOV rates are still "surprisingly" high (particularly for OOD text). In fact, this is due to the relatively small size of the available news text from which the splits were derived. In the meantime, we observe significant increases in the normalized perplexities. The perplexities are presented in their normalized forms since test corpus and vocabulary size can change among different approaches. We also excluded OOV words in the test sets from perplexity computations, as different models have different OOV rates. The normalized perplexity for each model is computed using

$$PP^{*} = (PP)^{\frac{N}{N_b}} \qquad (2)$$

where PP^{*} is the normalized perplexity, N is the number of tokens used to calculate the standard perplexity PP (recall that OOV tokens are excluded from computations), and N_b is the number of words in the original text.

It is widely accepted that an increase in perplexity indicates an increase in the difficulty of speech recognition. However, its impact on the recognition error rate is difficult to predict. It is interesting to note that the splitting methods have comparable normalized perplexities. The data driven approach without counts has the smallest lexicon. This is due to the fact that the data driven approach does not have the notion of stem and suffix, and can discover and exploit patterns across stems. This is not an advantage if one intends to use the approach as a basis for morphological analysis (recall its lower precision indicated in Table 2). However, it is an advantage for speech recognition since it speeds up the decoding, thanks to the small size of the lexicon. The other disadvantage is the shrinkage of LM span which has probably resulted in an increase in the normalized perplexity.

The word error rates (with/without adaptation) presented in Table 5 clearly indicate the improvement with word splitting. The best results are obtained using the data driven word splitting using counts. Despite its higher perplexity, lower LM quality, and higher OOV rate as compared to the morphological word splitting, its performance is significantly better. We think the win is due to its relatively longer lexical units. It should be noted that the compounding after splitting provides some further gain. We applied compounding only to the data driven splitting without counts, since it had the largest room for the lexicon growth. We iterated compounding two times. The threshold is set such that the lexicon size has grown approximately by 10,000 lexical units at each iteration.

6. Discussion

We have explored several methods of creating a lexicon for the Turkish LVCSR which are also applicable to other agglutinative languages. We have presented promising and encouraging results for the data driven approaches. The data-driven splitting and recompounding promises a very useful procedure for lexicon design. We are still far from a performance that is necessary for open domain practical applications. This is partly due to the relatively small text and audio corpora that we currently have for Turkish and partly due to the challenges in language and acoustic modeling of agglutinative languages.

7. References

1. B. Pellom, K. Hacioglu, "Recent Improvements in the CU Sonic ASR System for Noisy Speech: The SPINE Task",*IEEE ICASSP*, Hong Kong, 2003.
2. O. Salor, B. L. Pellom, T. Ciloglu, K. Hacioglu, M. Demirekler, "On Developing New Text and Audio Corpora and Speech Recognition Tools for the Turkish Language", *ICSLP'2002*, Denver, Colorado.
3. M. Akbacak, O. Salor, CSCI-6302 class project, University of Colorado at Boulder, USA, December, 2002.
4. Creutz, M and Lagus, K. "Unsupervised discovery of morphemes" In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pages 21-30, Philadelphia, PA, July 2002.
5. K. Oflazer, "Two-level description of Turkish Morphology", *Literary and Linguistic Computing*, Vol. 9, No. 2, 1994.
6. Dilek Zeynep Hakkani-Tur. Statistical Language Modeling for Agglutinative Languages. Ph.D. Thesis, Department of Computer Engineering, Bilkent University, August 2000, Ankara, Turkey.
7. E. Mengusoglu, O. Derro, "Turkish LVCSR: Database preparation and Language Modeling for an Agglutinative Language" in *ICASSP'2001*, Student Forum, Salt-Lake City, May 2001.
8. H. Dutagaci, L. M. Arslan, "A Comparison of Four Language Models for Large Vocabulary Turkish Speech Recognition," *ICSLP'2002*, Denver, Colorado.
9. K. Carki, Geutner, P. and Schultz, T., "Turkish LVCSR: Towards Better Speech Recognition for Agglutinative Languages," in *ICASSP'2000*, Istanbul, Turkey.
10. Vesa Siivola, Teemu Hirsimaki, Mathias Creutz, and Mikko Kurimo, "Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner " submitted to Eurospeech 2003.
11. C. Beaujard, M. Jardino, "Language Modeling Based on Automatic Word Concatenations", in *Proc. Eurospeech '99*, Budapest, Hungary, 1999.
12. G. Saon, M. Padmanabhan, "Data-Driven Approach to Designing Compound Words for Continuous Speech recognition," *IEEE Transactions on Speech and Audio Processing*, Vol.9, No.4, May 2001.