FINITE-STATE TRANSDUCER BASED MODELING OF MORPHOSYNTAX WITH APPLICATIONS TO HUNGARIAN LVCSR

Máté Szarvas mate@mateweb.net
Sadaoki Furui furui@cs.titech.ac.jp
Department of Computer Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo, 152-8552 Japan

ABSTRACT

This article introduces a novel approach to model morphosyntax in morpheme unit based speech recognizers. The proposed method is evaluated in our recent Hungarian large vocabulary continuous speech recognition (LVCSR) system. The architecture of the recognition system is based on the weighted finite state transducer (WFST) paradigm. The task domain is the recognition of fluently read sentences selected from a major daily newspaper. The vocabulary units used in the system are morpheme based in order to provide sufficient coverage of the large number of word-forms resulting from affixation and compounding. Besides the standard morpheme N-gram language model we evaluate the novel *stochastic morphosyntactic language model* (SMLM) that describes the valid word-forms (morpheme combinations) of the language. Thanks to the flexible transducer-based architecture of the system the morphosyntactic component is integrated seamlessly with the basic modules with no need to modify the decoder itself. The proposed *stochastic morphosyntactic language model* decreases the error rate by 17.9% relatively compared to the baseline trigram system. The morpheme error rate of the best configuration is 14.75% in a 1350 morpheme Hungarian dictation task.

1. INTRODUCTION

Hungarian is a Finno-Ugric language spoken by about 15 million people mainly in Hungary and in the neighbouring countries. Similarly to the other members of the Finno-Ugric language family Hungarian is an agglutinating language, that is, it relies heavily on suffixes.

Speech research has a long tradition in Hungary and there exist several research and commercial systems both for speech synthesis and automatic speech recognition (ASR). Previous ASR research efforts have been limited, however, to command and control tasks that have a limited vocabulary. Besides the shortage of resources the main obstacle that delayed the beginning of Hungarian LVCSR research is the size of the vocabulary and the complexity of the morphology. The number of different word forms is in the range of hundreds of millions according to estimates by different linguists and the accurate modeling of this vocabulary is not easy even with morphological decomposition because the number of inflection classes is very large. The other difficulty, related to the vocabulary representation problem, is the accurate computational representation of pronunciation.

In our previous work [4, 5] we proposed methods for treating both of these problems but until now we had experimental results only about the pronunciation modeling method [6]. In Section 2 of this article we describe the architecture of our new weighted finite state transducer based recognition system that was designed to facilitate an efficient implementation of both the phonology and morphology modeling methods. Then we describe the details of our proposed stochastic morphosyntactic language model in Section 3 and describe the results of the experimental evaluation of the method in Section 4. Finally, we conclude the article in Section 5 with a summary and suggestions for future work.

2. SYSTEM OVERVIEW

The standard knowledge components in a state-of-the-art ASR system are the acoustic model, the pronunciation model and the N-gram language model. The usual practice is to represent each of these different types of knowledge in their specialized data structure and to use dedicated code in the decoder for combining and searching them. This practice has been motivated by the need for efficient implementations and perhaps also by the incremental development of the systems.

The price of this highly optimized implementation is, however, the loss of flexibility for adding new knowledge sources to the system. The reason is that the specialized code gets increasingly complex and usually only the original developer of the decoder module would be able to add the new components. It has been widely understood for a long while that all the usual knowledge sources (KS) are just different instantiations of the same basic mathematical data structure: weighted finite-state transducers (WFSTs). But it has only recently been demonstrated [2] that a recognition system using this uniform data representation and generic algorithms for all KSs can achieve, with affordable system resources, a performance similar to and surpassing specialized systems.

This weighted finite-state transducer (WFST) based architecture [1, 2] is especially attractive for us because all the phonological and morphological dependencies described in [4, 5, 6] can be easily converted into a WFST representation. Moreover, we believe that higher level linguistic dependencies, such as the agreement of the number and person of the subject and the predicate, can also be represented in this framework. Therefore we designed our recognition system from the beginning according to the uniform-data WFST paradigm.

2.1. Review of WFST-based ASR

The main idea of WFST-based speech recognition [1, 2] is that each of the knowledge sources is represented as a weighted finite-state transducer. The search-space of the given task is obtained by combining the basic components using the composition operation of WFSTs. The main components of our current system besides the decoder itself are the acoustic model A, the context dependency mapping CD, the phonological rules P, the basic pronunciation dictionary D, the morphosyntactic rule set MS and the N-gram language model LM_N. A, CD, D and LM_N are standard components, P is introduced in [6], while MS is the main subject of this paper.

Using these components the recognition task is defined as finding the path with the highest likelihood in the integrated recognition network:

$$\text{ASR} = \text{best\_path}\ A \circ CD \circ P \circ D \circ MS \circ LM_N, \qquad (1)$$

where $\circ$ represents transducer composition.

3. LANGUAGE MODELING

As explained in the introduction, one of the difficulties in building a Hungarian LVCSR system is the modeling of the large vocabulary. For example, our language model (LM) database of 40 million words contains over 2 million different tokens before preprocessing. The number of different tokens remains over 1 million even after replacing all the number and punctuation characters with a white-space and converting all upper case characters to lower case. Therefore it is essential to use units smaller than words as the basic recognition unit in order to cover the vocabulary using system resources within practical limits. In most, if not all, languages words are built from smaller meaningful units: morphemes. Unlike word units, the number of morphemes is quite limited and they have been used with succes in speech recognition systems for different languages that suffer from the vocabulary size problem [3]. After the words are split up for morphemes, these systems are using morpheme N-gram language modeles instead of a word N-gram model.

Though the use of morpheme units proves to be quite effective for reducing the number of different recognition units, it has negative effects as well. Morphemes, especially pre- and suffixes, tend to be very short and acoustically confusable, leading to a high error rate for these units. Analizing the recognition errors reveals, however, that many errors result in a morpheme sequence that is not permitted by the language. Examples include attaching a verb suffix to a nominal stem, such as "destructive*[Adj] -ing[Grnd]*", or combining suffixes that cannot follow each other, such as "*-ing[Grnd] -ed[Past]*" (instead of the German name "*Inge*"). Due to the larger number of suffixes in agglutinating languages, the number of invalid combinations is much higher.

Such errors are easy to detect in the recognition output using a morphosyntactic rule-set, and one approach for improving performance could be to generate N-best lists and select the alternative that includes the smallest number of morphosyntactic errors. It is not guaranteed, however, that a good candidate would be included in the list for reasonable values of N. Therefore it is desirable to use the rules directly in the first pass of the recognition. This eliminates all invalid combinations, decreasing the error rate and potentially increasing the recognition speed.

In the next part of this section we describe the whole-word and the morpheme coverage statistics of our language-model training data to demonstrate the effectiveness of the morpheme based approach in reducing the vocabulary size. Then we review the standard morpheme N-gram language modeling approach in more detail as it is the baseline used in most current morpheme-based systems. Finally, we propose a new WFST-based method to integrate a morphosyntactic grammar with the morpheme N-gram model in a single recognition pass.

3.1. Training database and coverage

We used 40 months of text of a large Hungarian daily newspaper ("Magyar Hírlap") as the LM development data. There are 38.9 million white space separated tokens in the unprocessed database and the whole data size is 300 MByte. After normalizing the database by removing punctuation characters and splitting words into their constituent morphemes the total number of morpheme tokens is 74.1 million. We removed all digit characters and converted all upper-case characters to lower-case for the coverage tests that we conducted in order to assess the effectiveness of morpheme analysis in reducing the number of token types. The results of these tests are displayed in Figure 1. It is clear from the figure that the analysis significantly decreased the number of units necessary for a given coverage. For example to attain a coverage of 99% (OOV=1%) we need only 28k morpheme units while the number of necessary word units would be over 750k. Because the distance between the two curves is approximately constant and the "vocabulary size" axis is on a logarithmic scale, the size of the morpheme vocabulary is a constant factor (≈ 100) times smaller than the size of the whole word vocabulary for any given OOV-rate under 60%.

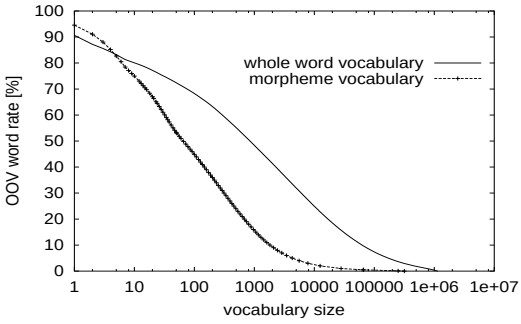

Fig. 1. Out of vocabulary word rate as a function of the vocabulary size when using whole word units and morpheme units.

Table 1. Language model perplexities for different vocabulary sizes (morpheme units, PP=perplexity).

Vocabulary size	PP 2GR	PP 3GR
1k	74.2	44.7
5k	85.6	49.1
10k	89.5	51.2
20k	97.9	52.9
65k	95.1	54.5

Finally, we note that the tail of the curve for morpheme coverage was generated by inflected words that our analyzer could not split, therefore the coverage would be much better for morpheme vocabulary sizes over 20k if we had a wider coverage analyzer.

3.2. The morpheme N-gram model

The standard approach to morpheme unit based language modeling is to use a morpheme N-gram model. In the first step all the words in the language model (LM) training data are split up to their constituent morphemes. In the second step an N-gram model is estimated using the morpheme sequence instead of the original word sequence.

In this approach the permitted word-forms (morpheme combinations) are represented implicitly by the transition likelihoods of the N-gram model. The advantage of this method is that it is easy to use because only a morpheme analyzer is needed to split up the words of the training data and the LM is estimated automatically. Even though an accurate morpheme analyzer is not available for all languages, a simple stem- and suffix-list based method may be equally suitable if it provides a consistent analysis.

We developed 2- and 3-gram models for different vocabulary sizes using this method. The perplexities of the models are displayed in Table 1. It is clear from the table that 3-gram models have a significantly smaller perplexity and their use is mandatory for high-accuracy recognition. The perplexity figures in themselves can be considered relatively good (small), but it is important to note that these are per-morpheme perplexities and not per-word perplexities. The per-morpheme perplexity is about half of the equivalent per-word perplexity because each word is split up to two morphemes on average. Besides, the resulting morphemes are much shorter than the original words and are therefore more confusable acoustically.

One disadvantage of this model is that all practical LM estimation algorithms have to apply a smoothing mechanism to avoid assigning zero likelihoods to valid but unseen word or morpheme combinations. Smoothing algorithms, however, cannot distinguish between inflected word-forms missing due to data scarcity and be-

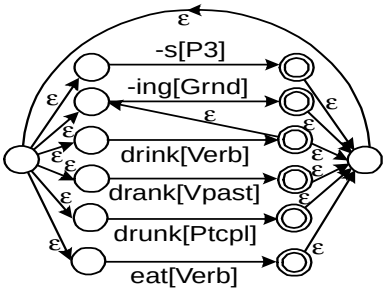

Fig. 2. Representation of inflected word-forms by the back-off bigram language model, LM_N.

tween inflected forms prohibited by the rules of the language. As a result, the smoothed language model assigns a positive likelihood to inflected forms that are not permitted by the rules of the language, for example the incorrect "érez-*tál*" in Hungarian or "apples-*ed*" in English.

3.3. The stochastic morphosyntactic language model

The accurate representation of the vocabulary of a language is also crucial in morphological analyzers. The most important component of a morpheme analizer is the morphosyntactic grammar that represents the permitted combinations of morphemes. A morphosyntactic grammar is frequently implemented as a finite-state automaton (FSA). A simplified example of a morphosyntactic grammar in the form of a finite-state automaton is represented in Figure 3. As opposed to the N-gram model example of Figure 2, this model does not permit ungrammatical sequences such as "*drank* [Verb+Past]-s[Present+Pers3]-*ing*[Geround]."

This representation is suitable for use in applications where the input is a deterministic character sequence. In speech recognition, however, the input is ambiguous and it is not enough to decide if a particular input sequence is valid or not, but we need to assign likelihoods to different input sequences.

In this regard the stochastic N-gram model and the deterministic morphosyntactic grammar are complementary. The smoothed N-gram model can assign a likelihood to any input sequence, but cannot distinguish between permitted and invalid morpheme sequenes. The morphosyntactic grammar, on the other hand, can only decide if a sequence is permitted or not, but it cannot assign a likelihood to permitted sequences. We would like to have a language model that is accepting exactly those sequences that the morphosyntactic grammar is accepting and to the accepted sequences it assigns the same likelihood as the N-gram model. The finite-state intersection $MS \cap LM_N$ of the two FSA-s has exactly this property: by definition, the finite state intersection of two weighted FSA-s is accepting those sequences that both automata accept. But LM_N is accepting all sequenes, therefore $MS \cap LM_N$ is accepting the same set as MS. And the intersection of two weighted FSA-s assigns the sum of the original weights to any accepted sequence. But the unweighted MS assigns a weight of 0 to any sequence, therefore the sum is the weight assigned by LM_N.

Because the resulting language model, $MS \cap LM_N$, integrates the advantages of the stochastic N-gram model and the morphosyntactic model we call it the *stochastic morphosyntactic language model* (SMLM). The SMLM resulting from the intersection of the N-gram model LM_N in Figure 2 and the morphosyntactic grammar MS in Figure 3 is depicted in Figure 4. We can see in the figure, that MS eliminated the invalid combinations from LM_N while retaining the likelihoods of the valid transitions. The final step in making the SMLM a correct language model is to renormalize the weights on the transitions leaving each node because the intersection with MS is removing many transitions from

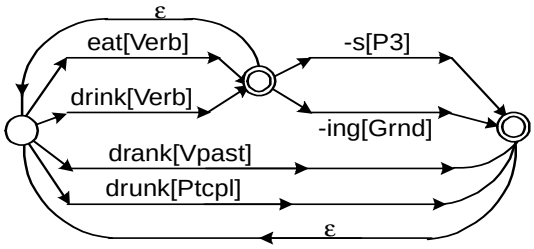

Fig. 3. Representation of inflected word-forms by the morphosyntactic grammar, MS.

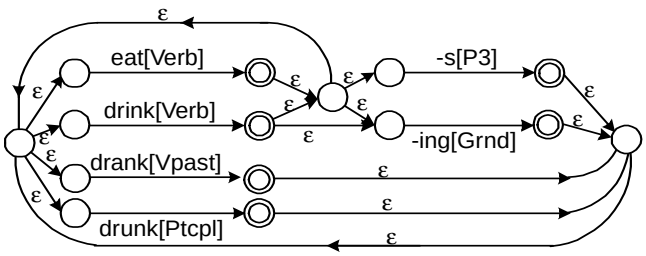

Fig. 4. Representation of inflected word-forms by the stochastic morphosyntactic language model, $SMLM$. The morphosyntactic model filtered out the ungrammatical combinations at the price of increasing the size of the network.

LM_N and the sum of the weights becomes smaller than 1 for many nodes.

4. EXPERIMENTAL EVALUATION

We conducted continuous speech recognition experiments in order to evaluate the usefulness of the proposed stochastic morphosyntactic language model in a real task. The conditions of the experiments are described in detail in [6], therefore here we provide only a brief summary in the interest of saving space.

Conditions and results. The testing database contained read newspaper sentences from the same Hungarian newspaper that was used for training the LM-model (with no overlap between testing data and LM-training data). For computational reasons, for these experiments we used only those sentences from the database that could be covered with 1350 morphemes (closed vocabulary recognition).

The acoustic models used in the experiments were speaker and gender independent triphone HMMs trained with 1 hour of read speech from 30 speakers. The feature parameters were 13 MFCC parameters plus their first and second order derivatives. The phoneme recognition error rate using these models was 39.13%, indicating severe under-training of the acoustic models. The decoder used was a simple frame synchronous Viterbi decoder. The pronunciation dictionary was generated automatically as described in [6], but the phonology modeling component was not used.

We precompiled recognition networks using 4 different language models: a bigram and a trigram model to serve as the baseline, and the corresponding stochastic morphosyntactic models for both cases. The final normalization step described at the end of Section 3.3 was not applied in the case of the SMLM-s. The size of the 4 networks is displayed in Table 2. We can see that the application of the morphosyntactic model, MS, increased the size of both networks by more than a factor of 2. The reason is that the composition of the MS model introduced several new back-

Table 2. Number of arcs in the precompiled recognition network for different language models (k=thousand).

	N-gram LM		SMLM
Bi-gram	582 k	$\rightarrow$	1144 k
Tri-gram	1085 k	$\rightarrow$	2504 k

Table 3. Comparision of speaker independent morpheme error rates with different language models.

	N-gram LM		SMLM
Bi-gram	21.20%	$\rightarrow$	18.89% (-10.9%, rel.)
Tri-gram	17.97%	$\rightarrow$	14.75% (-17.9%, rel.)

off nodes like the starting node of the two suffixes in the network of Figure 4.

The recognition error rates[1] for the 4 cases are displayed in Table 3. The application of the morphosyntactic grammar decreased the error rate of the bigram model by about 11% relatively. The use of a trigram LM decreased the error rate even more with a slightly smaller network size. However, the improvement provided by trigrams was completely independent from the improvement by morphosyntax, since the use of MS together with the trigram model gave an additional 18% relative improvement, larger than in the case of the bigram model. This is similar to the findings in [6] in that a sophisticated modeling technique may give more relative improvement when the baseline system is better.

Result analysis. The result of the error-rate reduction analysis is summarized in Table 4. The first observation is that the use of morphosyntax could not reduce the number of deletion errors. The reason is that almost all deletion errors are the result of missing syllables due to fast speech. Missing syllables cause a very strong acoustic mismatch and probably they can be compensated only by direct modeling in the pronunciation model.

21.4% of the eliminated errors was an insertion error. Most of this reduction is due to the elimination of superfluous short suffix morphemes frequently inserted during the leading and closing silence period. These morphemes are frequently inserted by the recognizer when there is some non-speech noise during the silence periods, but usually the short morpheme that well matches the acoustic signal is not permitted in that position. The other source of reduction in the number of insertion errors is the elimination of many "splitting errors." Splitting errors are those errors where a longer, usually content-, morpheme is split up for two or more short, usually suffix-, morphemes. The result of these splits is usually ungrammatical, either because the first morpheme cannot be connected to the preceding word, or because the two morphemes cannot be connected with each other. Elimination of such errors reduces both the number of insertion and substition errors because one of the members of the resulting split causes a substitution error, while the rest are causing insertion errors.

Finally, the largest decrease is in the number of substitution errors. There are three sources of this decrease. One source is from the eliminated split-errors we described above. The other sources are mistaken stems restored by the acoustically well matching suffix and suffixes restored by the stem. Sometimes the connection constraint of two morphemes can eliminate a sequence of several substitution errors.

[1] defined as $100\frac{S+D+I}{N}$%, where S, D and I denotes the number of substitutions, deletions and insertions and N denotes the total number of morphemes in the test-set

Table 4. Distribution of error rate reduction between deletion, insertion and substitution errors.

Error type	deletion	insertion	substitution
Reduction	0%	21.4%	78.6%

5. CONCLUSION AND FUTURE WORK

In this paper we introduced a novel stochastic morphosyntactic language model that integrates the advantages of N-gram models and morphosyntactic grammars in morpheme-unit based speech recognizers. The main difference of our model to the standard N-gram model is that our model can utilize the strong connection constraints between different morpheme classes, whereas the standard (smoothed) N-gram model permits any combination. Thanks to the flexible finite state transducer based design of our recognizer, the morphosyntactic module could be integrated into the system with no need to modifiy the decoder itself. We evaluated the method in a Hungarian dictation task and the use of the morphological grammar reduced the baseline N-gram based morpheme error rate by about 11% in the bigram and by about 18% in the trigram case. The source of the improvement is the elimination of insertion and substitution errors, but the method was not effective for reducing the number of deletion errors. The cost of the improved accuracy is the doubling of the decoding-network size. This might be alleviated by adopting a two-pass strategy, though the two-pass method does not guarantee the same improvement as the single-pass method. Although these results were obtained in a small task, we believe that it will scale well for larger vocabularies because the problem that the method is addressing is getting more acute with the increase of vocabulary size. Furthermore, we believe that the method is applicable to other languages as well because all recognizers that must use morpheme units suffer from these problems.

Further improvement of the SMLM could be expected from properly normalizing the model as described at the end of Subsection 3.3. The acoustic modeling component in the system could be improved by modeling phoneme duration explicitly because the 25 long consonants differ exclusively in duration from their short counterparts and without duration modeling the current system is unable to distinguish these pairs. Finally, we expect further improvement of the recognition accuracy from combining the SMLM with the phonology modeling method introduced in [6].

6. REFERENCES

[1] M. Mohri. Finite State Transducers in Language and Speech Processing. *Computational Linguistics*, 23:2, 1997.

[2] M. Mohri, F. Pereira, M. Riley. Weighted Finite-State Transducers in Speech Recognition. In *Proc. ISCA Automatic Speech Recognition 2000.*, pp. 97–106.

[3] K. Ohtsuki, T. Matsuoka, T. Mori, K. Yoshida, Y. Taguchi, S. Furui, K. Shirai. Japanese large-vocabulary continuous-speech recognition using a newspaper corpus and broadcast news. *Speech Communication (28):155–166, 1999.*

[4] M. Szarvas, T. Fegyó, P. Mihajlik, P. Tatai. Automatic Recognition of Hungarian: Theory and Practice. *International Journal of Speech Technology*, 3(3/4):237–251, 2000.

[5] M. Szarvas, S. Furui. The use of finite-state transducers for modeling phonological and morphological constraints in automatic speech recognition. In *Proc. Autumn Meeting of the Acoustical Society of Japan.*, 2-1-20, pp. 87–88, 2001.

[6] M. Szarvas, S. Furui. Finite-state transducer based Hungarian LVCSR with explicit modeling of phonological changes. In *Proc. ICSLP 2002.*, pp. 1297–1300.