Evaluation of the Stochastic Morphosyntactic Language Model on a One Million Word Hungarian Dictation Task

Máté Szarvas   *Sadaoki Furui*
mate@mateweb.net   furui@cs.titech.ac.jp

Department of Computer Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo, 152-8552 Japan

Abstract

In this article we evaluate our stochastic morphosyntactic language model (SMLM) on a Hungarian newspaper dictation task that requires modeling over 1 million different word forms. The proposed method is based on the use of morphemes as the basic recognition units and the combination of a morpheme N-gram model and a morphosyntactic language model. The architecture of the recognition system is based on the weighted finite-state transducer (WFST) paradigm. Thanks to the flexible transducer-based architecture, the morphosyntactic component is integrated seamlessly with the basic modules with no need to modify the decoder itself. We compare the phoneme, morpheme, and word error-rates as well as the sizes of the recognition networks in two configurations. In one configuration we use only the N-gram model while in the other we use the combined model. The proposed stochastic morphosyntactic language model decreases the morpheme error rate by between 1.7 and 7.2% relatively when compared to the baseline trigram system. The morpheme error-rate of the best configuration is 18% and the best word error-rate is 22.3%.

1. Introduction

The modeling of the large number of different word-forms resulting from inflection and derivation is one of the obstacles that delays the development of wide coverage large vocabulary speech recognition systems for many languages. In our previous work [4] we have proposed a method that is based on the use of morpheme units and the combination of the statistical N-gram model with a finite-state morphosyntactical grammar. The results in [4] suggested that the resulting stochastic morphosyntactic language model (SMLM) is quite effective in reducing the morpheme-errors in our Hungarian dictation system. Those results, however, were based on a limited 1500 word dictation task and a small test set. In order to confirm the effectiveness of the method in larger problems, we extended our recognizer to use a 25,000 morpheme inventory that can provide a 99% coverage of the more than 1 million word forms present in our newspaper corpus. The other major improvement compared to the system in [4] is that our new system can combine the morphemes to produce whole words as the recognition output, while the old system was outputting each morpheme in isolation.

In the remaining part of this section we give a brief summary of weighted finite-state transducer (WFST) based speech recognition because our system is relying on this paradigm. In Section 2 we describe our language model data and review the stochastic morphosyntactic language model originally introduced in [4]. Then we provide the details of the experimental evaluation in Section 3 and conclude the article in Section 4 with a summary and suggestions for future work.

Summary of WFST based speech recognition. It has been understood for a long while that each of the standard knowledge sources (KS-s) in automatic speech recognition (ASR) systems are just different examples of the same mathematical data structure: weighted finite state transducers (WFST-s). Moreover, it is also possible to represent many "non-standard" KS-s in the form of a WFST. Two examples are the use of phonological [3] and morphological [4] rules. In order to be able to use different KS-s flexibly, we designed our system from the beginning according to the uniform-data WFST paradigm [2]. In this paradigm each KS is represented as a WFST and the search space of the recognition task is obtained by combining the basic components by the the composition operation of WFST-s [2]. The main components of our 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, while P is introduced in [3] and MS is introduced in [4].

2. 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 success in speech recognition systems for different languages that suffer from the vocabulary size problem. After the words are split up for morphemes, these systems are using morpheme N-gram language models 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 "de-structive[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
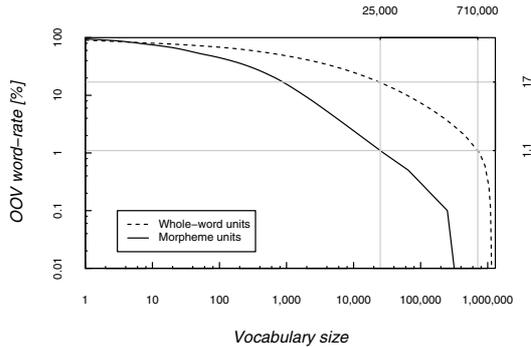
Figure 1: Out of vocabulary word rate as a function of the vocabulary size when using whole word units and morpheme units.

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

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

Figure 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.

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.

2.1. 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 we can attain a coverage of 89.9% (OOV=1.1%) with 25,000 morpheme units while the number of necessary word units would be 710,000, more than 28 times larger. 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 (≈ 30) times smaller than the size of the whole word vocabulary for any given OOV-rate under 10%. Finally, we note that the tail of the curve for morpheme coverage was generated by inflected words that our analyzer failed to split, therefore the coverage would be much better for morpheme vocabulary sizes over 20k if we had a wider coverage analyzer.

2.2. 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.

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
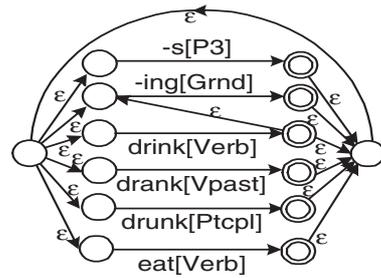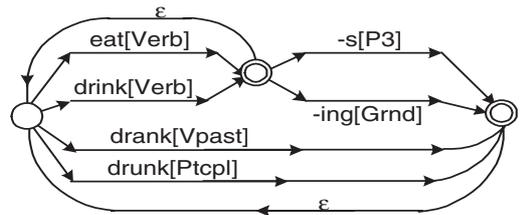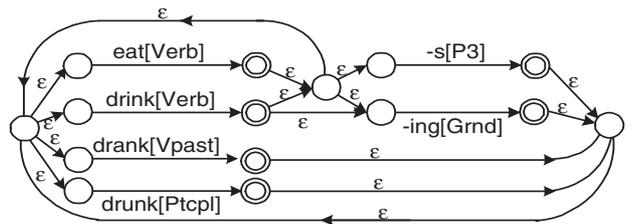
data scarcity and inflected forms prohibited by the rules of the language. As a result, the smoothed language model assigns a positive likelihood to inflected forms that are not permitted by the rules of the language, for example the incorrect "*drank* [Verb+Past]-*s*[Present+Pers3]" in Figure 2.

2.3. The stochastic morphosyntactic language model

The permitted morpheme combinations of a language are described by its morphosyntax. Unlike sentence-syntax, morphosyntax can be efficiently implemented as a finite-state automaton (FSA) for most natural languages. 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 valid sequences.

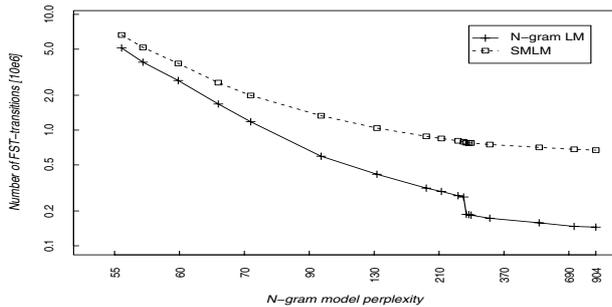In this regard the stochastic N-gram model and the deterministic morphosyntactic grammar are complementary. The

Figure 5: The size of the precompiled recognition network as a function of the base-line LM_N-perplexity for the N-gram and the $SMLM$ language model. (The transducers were determinized but they were not factorized, therefore the input side of each transition corresponds to one phoneme.)

smoothed N-gram model can assign a likelihood to any input sequence, but cannot distinguish between permitted and invalid morpheme sequences. 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 sequences, 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 LM_N and the sum of the weights becomes smaller than 1 for many nodes.

3. 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 [3], 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). There are 20 speakers' voice in the database comprising a total of 613 sentences.

The acoustic models used in the experiments were speaker and gender independent triphone HMMs trained with 1 hour of read speech from 30 speakers (different from the testing 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 [3], but

the phonology modeling component was not used.

We precompiled two sets of recognition networks using different language models. The networks in the first set were based on N-gram models with different perplexities to serve as the baseline. The members of the second set were the corresponding SMLM-s obtained by intersecting the members of the first set with the morphosyntactic grammar. The final normalization step described at the end of Section 2.3 was not applied in the case of the SMLM-s. The size of the networks is displayed in Figure 5. (The source of the discontinuity in the figure is unclear, but we double-checked the measurements and found no error.)

We can see that the application of the morphosyntactic model, MS, increased the size of the baseline models in each case. The relative increase is, however, dependent on the perplexity. For the smaller baseline models (larger perplexity) the relative increase is high. For the smallest network the increase is more than 4.5 times (from 145,000 to 671,000) because the baseline is a unigram LM with a small number of transitions, but the SMLM has to encode all the morphological dependencies. At larger sizes the relative increase is much smaller. For example the size of the best N-gram model increased only by 30% (from 5.12 million transitions to 6.63 million). This is probalby because the baseline model already encoded much of the morphological information extracted from the training data.

The letter, morpheme and word recognition error-rates[1] are displayed in Figures 6–8. The application of the morphosyntactic grammar decreased the error-rate in all cases, though not to the same extent. All error-rate reductions were found, however, to be significant above the 99.5% level using the matched pairs sentence word error test of [1]. The relative error rate reductions range between 0.77 and 3.22% for the letter error-rates, between 1.67 and 7.22% for the morpheme error-rates and 1.85 and 5.72% for the word error-rates, with larger relative improvements obtained for higher perplexities in general, though not strictly monotonically. The word error rate of the best baseline system was reduced from 22.74% to 22.32%, corresponding to a 1.85% relative decrease.

Result analysis. The result of the error-rate reduction analysis is summarized in Figure 9. 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.

The contribution of insertion error reduction to the total error reduction is 37.9%. 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.
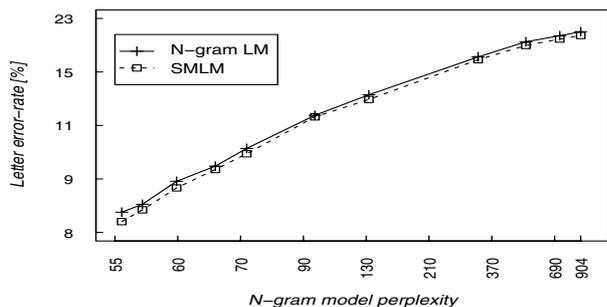
Figure 6: Letter error-rate as a function of the base-line LM_N-perplexity for the N-gram and the $SMLM$ language model.
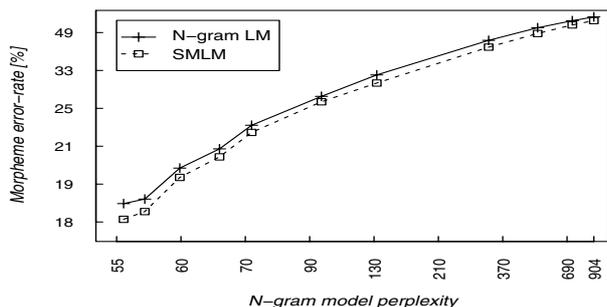

Figure 7: Morpheme error-rate as a function of the base-line LM_N-perplexity for the N-gram and the $SMLM$ language model.
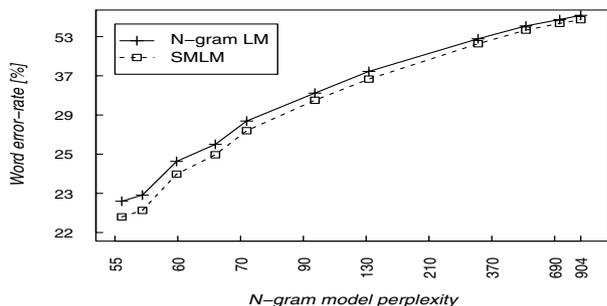

Figure 8: Word error-rate as a function of the base-line LM_N-perplexity for the N-gram and the $SMLM$ language model.

Figure 10 displays the distribution of the per-sentence error change. We can see that the use of the stronger model actually increased the number of errors in some of the sentences. Checking the source of this increse, we found that it was always caused by strong acoustic mismatch, such as a missing syllable. In such cases the "softer" N-gram model could recover more quickly by traversing an ungrammatical morpheme sequence, while the SMLM introduced a longer sequence of errors in order to maintain grammaticality.

4. Conclusion and future work

In this article we described the experimental evaluation of our stochastic morphosyntactic language model on a Hungarian dictation task that requires the modelling of over 1 million different word forms. We found that the use of morpheme connectivity constraints can provide significant error rate reductions on this larger task as well, though the reductions are in general smaller than in our previous experiments [4]. Moreover, we evaluated the contribution of the morphosyntactic compo-
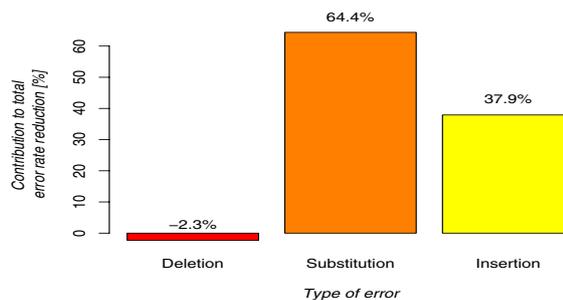

Figure 9: The contribution of different error-types to the total error rate reduction. The contribution of deletion errors is negative because the number of deletion errors increased.
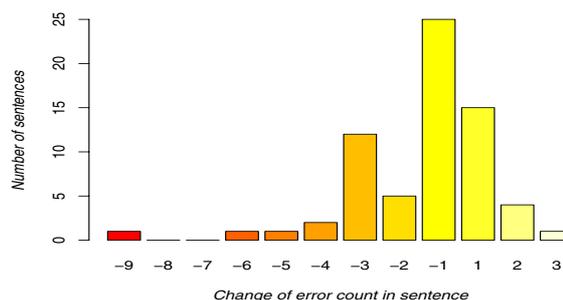

Figure 10: The distribution of error-count change in the test sentences. (The total number of sentences was 613. The "no-change" case is not depicted in the figure.)

nent over a wide range of baseline N-gram model perplexities. The general observation is that the morphosyntactic component gives more improvement when the baseline N-gram model has higher perplexity, though it decreased the error rate even in the case of the best N-gram model. A new finding compared to our previous results is that the increase in the size of the combined model is dependent on the size of the baseline model and the relative increase is much smaller for larger (or "stronger") baseline models. This is important, because a 30% size-increase is tolerable even for the largest models, but a 4-times increase would probably be not acceptable considering the potential gains.

Further improvement of the SMLM could be expected from properly normalizing the model as described at the end of Subsection 2.3. 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 [3].

5. References

[1] L. Gillick and S. Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *ICASSP 89*, pp. 532–535.

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

[4] M. Szarvas, S. Furui. Finite-state transducer based modeling of morphosyntax with applications to Hungarian LVCSR. To appear in *Proc. ICASSP 2003*.