Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner

Vesa Siivola, Teemu Hirsimäki, Mathias Creutz, Mikko Kurimo

Neural Networks Research Centre
Helsinki University of Technology
Vesa.Siivola@hut.fi, Teemu.Hirsimaki@hut.fi, Mathias.Creutz@hut.fi, Mikko.Kurimo@hut.fi

Abstract

We study continuous speech recognition based on sub-word units found in an unsupervised fashion. For agglutinative languages like Finnish, traditional word-based n-gram language modeling does not work well due to the huge number of different word forms. We use a method based on the Minimum Description Length principle to split words statistically into sub-word units allowing efficient language modeling and unlimited vocabulary. The perplexity and speech recognition experiments on Finnish speech data show that the resulting model outperforms both word and syllable based trigram models. Compared to the word trigram model, the out-of-vocabulary rate is reduced from 20% to 0% and the word error rate from 56% to 32%.

1. Introduction

Word n-gram models are the most common statistical language models used in speech recognition today. For languages with rather simple morphology, such as English, the coverage of a reasonably sized vocabulary is sufficient for general recognition. However, for agglutinative languages, such as Finnish, a vocabulary containing the 65 000 most frequent word forms typically excludes a considerable part of the words in test sets.

The difference is explained by properties of the languages. In English, compound words are usually written apart, and the syntactic role or meaning of a main word can be modified using short words, such as prepositions and articles, e.g., "in", "from", and "the". In Finnish, the number of distinct word forms is much higher, as compound words are written together and lengthy sequences of suffixes indicating, e.g., case, number and person can be appended to the word stem. For example, the word "Tietä+isi+mme+kö+hän?" could be translated as "Would we really know?"

In order to increase the effective coverage of the vocabulary, a lexicon containing smaller units than entire words can be constructed. Ideally, any word form can then be obtained through a concatenation of suitable sub-word units. However, it is challenging to find a set of units for which a good language model can be created.

In [1, 2, 3] morphological rules are used to produce sub-word units. The rules are based on language-dependent prior assumptions about stems and suffixes. In [1], two additional methods are tested. In one of the methods, the coverage of sub-word units is maximized given the lexicon size. In the other, the most common word forms are taken as a basis and the rest of the words are generated using a combination of the most common words and sub-word units.

In this paper, we utilize an entirely unsupervised algorithm presented in [4], which discovers sub-word units from a text corpus. The algorithm is language-independent and simply assumes that words consist of sequences of segments. No distinction is made between different categories of segments, such as stems, prefixes, or suffixes. We call the segments *morphs* because they resemble actual morphemes of the language. The optimal set of morphs is obtained by optimizing a cost criterion derived from the Minimum Description Length (MDL) principle [5]. Experiments show that morphs can be used in language models yielding a practically unlimited vocabulary. Speech recognition experiments confirm a considerable drop in word error rates when compared to language models based on words or syllables.

2. Language modeling

Traditionally, lexicons used in speech recognition contain the most frequent words as lexical units, and the language model is used to assign probabilities for word sequences. Another approach is to build the lexicon out of shorter units, and allow the language model to concatenate the units, generating a much larger, possibly infinite, vocabulary.

In this paper, we compare three lexicons with different lexical units: words, syllables, and morphs found by an unsupervised splitting algorithm. Throughout the paper, we use the general term *token*, where any of the lexical units (word, morph or syllable) is applicable.

In many languages, the pronunciation of a word (e.g., the phoneme sequence) can be roughly derived from the written form. This is the case for Finnish, where almost each letter corresponds to one phoneme. However, a distinction is made between short and long sounds and the latter are spelled with double letters, e.g., "tuli" and "tuuli" ("fire" and "wind"). In this work, we treat long and short sounds as separate phonemes and do not allow any token boundary in the middle of a phoneme.

Because very large lexicons make the speech recognition process time and memory consuming, we have restricted the size of our lexicons to approximately 65 000 tokens.

2.1. Words

Using words as lexical units is straightforward. The most frequent word forms are selected from a training corpus and put in the lexicon. However, due to the practical size limit of the lexicon, the word-based lexicon cannot cover the Finnish language very well.

2.2. Syllables

A good segmentation of Finnish words into syllables is possible using a reasonably simple ruleset. Our syllable lexicon can gen-

erate practically all words of Finnish origin, but it misses some foreign names.

2.3. Morphs

Morphemes are the smallest meaning-bearing elements of language. As any word form can be constructed by a combination of morphemes, morphemes seem appropriate lexical units in huge-vocabulary speech recognition. In this work, we use an unsupervised algorithm that discovers segments that bear resemblance to the actual morphemes of the language. We call these segments morphs. The algorithm was originally presented as the *Recursive MDL* method in [4].[1]

2.3.1. MDL model

The algorithm learns a set of morphs from a text corpus used as training data. The task is to rewrite the words in the corpus as sequences of morphs. Every morph discovered is added to a morph lexicon. The optimal segmentation of the corpus into morphs is such that the representation of the segmented corpus together with the representation of the morph lexicon is as compact as possible. This corresponds to the MDL (Minimum Description Length) principle [5].

The optimization criterion can be written as a two-part cost function C:

$$\begin{aligned} C &= \text{Cost(Corpus)} + \text{Cost(Lexicon)} \\ &= \sum_{\text{occurrences}} -\log_2 p(m_i) + \sum_{\text{morphs}} k \cdot l(m_j)\,. \end{aligned} \quad (1)$$

The cost of the corpus is a sum over all morph occurrences in the corpus. The idea is that the corpus is rewritten as a sequence of morphs m_i and each morph is replaced by a morph pointer. The cost—or code length in bits—of each pointer is determined by the probability of the morph, $p(m_i)$, computed as the maximum likelihood estimate of the morph in the corpus. Thus, frequent morphs receive short pointers, whereas rare morphs receive long pointers. The length of the pointer in bits is obtained by taking the negative base-2 logarithm of the probability: $-\log_2 p(m_i)$.

The cost of the lexicon consists of the code length of all distinct morphs m_j spelled out. It takes $k \cdot l(m_j)$ bits to code one morph, where $l(m_j)$ is the length in phonemes of the morph. The number of bits required to code one phoneme is k. In this work, the value of k is set to $\log_2 60$, since there are 60 different phonemes[2] in use and we have selected a uniform code length for every phoneme. Alternatively, the fact that different phonemes have different probabilities could have been taken into account. However, we have tried such a model and it did not improve performance.

2.3.2. Search for the optimal model

Finding the set of morphs that minimizes the cost function given a particular corpus is a non-trivial task. We use the recursive segmentation algorithm from [4], with the exception that the search does not take place incrementally, but in batch mode. All words in the corpus are shuffled, and for each word, every possible split into two parts is tested. The split location (or no split) yielding the lowest cost is selected, and in case of a split, the two parts are recursively split in two. The whole corpus is iteratively reprocessed until the total cost of the model converges. Figure 1 illustrates the hypothetical recursive splitting of two English words.

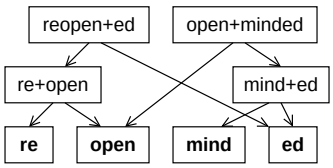

Figure 1: Hypothetical binary splitting trees for the words *reopened* and *openminded* (segmented as "re+open+ed" and "open+mind+ed"). The leaf nodes of the trees correspond to morphs discovered by the algorithm.

2.3.3. Lexicon pruning

As the number of distinct word forms in our Finnish corpus is very high, about 1.6 million, the number of morphs discovered by the algorithm turns out to be about 300 000, which exceeds the limit of 65 000. Therefore, we prune the morph lexicon to contain only the 65 000 most common morphs, after which we resegment the corpus using only these morphs. This means that rare word forms may be split into rather small segments. The pruned morph lexicon contains a token for each individual phoneme, so in the worst case any word can be rewitten as sequence of phonemes. Thus, the out-of-vocabulary rate remains at 0% regardless of the pruning of the lexicon.

2.4. Construction of the language models

For each of the three lexicons, a trigram language model was generated over the lexical units. The CMU language modeling toolkit [6] was used with Good-Turing smoothing and back-off to lower order n-grams.

Because the word lexicon contains whole words, a word break can be assumed after each token. With syllables and morphs, the word break has to be modeled explicitly. We have added a separate word break token to the syllable and morph lexicons, and it is treated as a normal token during the language model training.

3. The speech recognition system

3.1. Acoustics

The acoustic part of the recognizer is a traditional hidden Markov model with Gaussian mixture models. The mixture centers are initially placed with Self-Organizing Maps and this initial model is refined by Viterbi training [7]. Triphone models were trained using a simple back-off scheme: If there is sufficient data for training a triphone model, it is trained. Otherwise, a diphone model is used. If the data is still insufficient, a monophone model is trained. More elaborate ways of clustering the triphones were not studied, since the focus of this paper is in language modeling. Since our decoder does not take cross-token phoneme contexts into account, the triphone models are always truncated to diphones at word, morph or syllable boundaries. The corresponding truncations should be taken into account, when training the triphones. Ignoring them results in significantly worse performance.

In Finnish, the difference between the long and short variant of the same phoneme is mainly expressed by the duration. Our current acoustic model treats these variants as one and it is up to the language model to decide which variant will be used. We

[1] An online demo is available on the Internet at http://www.cis.hut.fi/projects/morpho/.

[2] Note that long and short sounds are treated as separate phonemes, which explains the high number.

still manage to get reasonably good recognition results, but this phenomenon needs to be more carefully modeled in the future.

3.2. The decoder

Our one-pass decoder is based on *time-synchronous search* and *start-synchronous trees* according to the terminology used in [8]. The main idea is to group *hypotheses*[3] ending at the same frame, and store them in frame-wise stacks. The benefits of this approach are that a single token expansion can be shared with all hypotheses ending at the same frame, and complex language models can be used quite easily. However, because hypotheses in a stack may have different phonological and language model contexts, it is hard to exploit cross-token triphones and language model probabilities in the acoustic search.

Figure 2 illustrates the decoding process. At each step, the decoder selects the earliest frame with a stack containing hypotheses. The acoustically most promising tokens starting from the corresponding frame and their ending times are found by Viterbi search. Each hypothesis in the stack is then expanded by the best tokens and placed in the appropriate stacks according to the best ending times within the search window. At this point, the language model probabilities are included.

In the figure, we can also see how the word breaks are handled with morphs and syllables. When a hypothesis is placed in a stack, the decoder creates another copy and appends an word break token to it. These word breaks are scored only by the language model if there is no explicit silence between the tokens. Continuous speech seldom has very clear acoustic information about word breaks. For simplicity, the figure shows only one ending time for each token but, in practice, several most promising ending times are used for each token, and hypotheses are created accordingly.

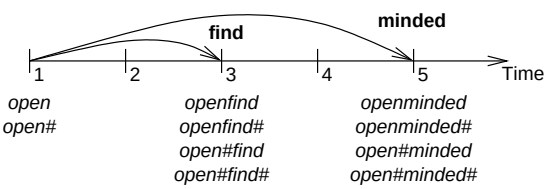

Figure 2: The decoder expands hypotheses with the acoustically best tokens, and places them in the appropriate stacks. Word breaks are marked as #.

4. Experiments

4.1. Text and speech data

The text data for training the morph segmentation and trigram language models was from two sources: short newswires from the Finnish News Agency (STT), and books, magazines and newspapers from the Finnish IT center for science (CSC). In total, the data contained 30 million words (1.6 million distinct word forms).

The speech data used for training the acoustic models was independent of the text data. A talking book read by a female speaker and the corresponding transcription was used. 12 hours of the book was used for training the acoustic models, 8 minutes for tuning the decoder parameters, and 28 minutes for testing.

[3] In this paper, hypotheses are token sequences with an ending time and cumulative log-probability.

lexical unit	number of units	OOV words	trigram hits	token perpl.	word perpl.
word	64000	20.2%	18.2%	-	4 300
morph	64684	0.00%	77.6%	84	28 500
syllable	36850	0.02%	98.9%	15	65 800

Table 1: Perplexity results. OOV = out-of-vocabulary words. *Trigram hits* shows the proportion of test set trigrams found in the model. Since the word break token is very common, the syllable and morph models get low token perplexities.

4.2. Perplexity

The perplexity measures how well a language model fits the test data. For a trigram word model, it is defined as

$$Perp(w_1, w_2, w_3, \ldots, w_W) = \left[P(w_1)P(w_2|w_1) \prod_{i=3}^{W} P(w_i|w_{i-1}, w_{i-2}) \right]^{-\frac{1}{W}} \quad (2)$$

Per-token perplexity for syllable and morph models is calculated by substituting the words with the tokens and summing over the number of tokens N instead of number of words W. However, since the word, syllable and morph models operate with completely different token sets, these measures are not comparable with each other.

To make the perplexity results comparable, we computed *word perplexity* for the morph and syllable models. It can be defined as

$$Perp(t_1, t_2, t_3, \ldots, t_N) = \left[P(t_1)P(t_2|t_1) \prod_{i=3}^{N} P(t_i|t_{i-1}, t_{i-2}) \right]^{-\frac{1}{W}} \quad (3)$$

where W is the number of word break symbols in the sequence of tokens $\{t_1, \ldots, t_N\}$. This measure is comparable across all of the models. The perplexity tests were run on the transcription of the acoustic training data (about 49 000 words) and the results are shown in Table 1.

The results indicate that the morph and syllable models cover the test set far better than the word model, and also give reasonable perplexities even though they can generate an infinite number of distinct word forms. The perplexities may sound high when compared to usual perplexities reported for English text (around 150 [9]), but we stress that a typical Finnish word corresponds to more than one English word, which naturally makes the perplexities higher. For example, if we assumed that one Finnish word would correspond to two English words on average, and the word perplexity of our morph model was computed over the double number of words, the result would be about 168 (square root of the original perplexity 28 500).

Also, the word model seems to get lower perplexity than the other models but the comparison is not fair. Because of the high OOV rate of the word model, about every fifth word is ignored in the perplexity computation. These words are the rarest ones, and taking them into account would increase the perplexity significantly.

4.3. Speech recognition

The speech recognition experiments were run with the following three trigram language models: the baseline word model,

lexical unit	WER	ToER	LER
word	56.4%	-	13.8%
morph	31.7%	20.3%	7.3%
syllable	43.9%	18.1%	10.9%

Table 2: Speech recognition results using trigram models.

syllable model and morph model. To make the results comparable, development data was used to tune the decoder parameters so that the real-time factor of the decoding process was around 30 for each model. Setting the real-time factor to be equal for all test runs was the only way to make the comparisons fair because different models required quite different pruning parameters for best performance. Note also, that our system is developed for easy prototyping and not for efficient memory use or real-time recognition.

The evaluation data was divided into 20 segments in order to measure the statistical significance of the results, and three error rates were computed for each segment: traditional word error rate (WER), token error rate (ToER) over the sub-word units, and letter error rate (LER).

Because a Finnish word corresponds to more than one English word on average, comparing the word error rates between the languages is not fair. The morph error rate in Finnish is perhaps closer to the word error rate in English, but also here, a direct comparison is impossible. Letter error rates may describe best the quality of the recognition result if the result has to be corrected manually but the proper error measure naturally depends on the application at hand. Phoneme error rates used in many papers depend highly on the size of the phonetic alphabet and are not discussed here. The results of our experiments are shown in Table 2. All pairwise differences between the models were statistically significant.

4.4. Discussion

The results show that the morph-based model outperforms the other models by a good margin. This is even more remarkable when noted that the decoder does not handle cross unit contexts, so the word tokens, being longer, can exploit the phonological contexts more. On the other hand, it can be argued that the comparison is not fair since the word model has a 20% out-of-vocabulary rate to start with. However, in order to reduce the out-of-vocabulary rate significantly, the vocabulary size should be increased by orders of magnitude, which would make the decoding computationally infeasible.

The benefit of word splitting is probably greatest in languages like Finnish, Hungarian and Turkish, which have rich morphology. However, the approach should also be applicable to less inflected languages, such as Czech [2], German [1], French and even English [3].

It would be interesting to compare our morphs to true linguistic morphemes. Unfortunately, there exist neither ready-made morpheme lexicons for Finnish, nor software for segmenting words into morphemes.

In future, we intend to explore higher order n-grams for all models. Whereas the trigram word model covers 3 words, the trigram syllable model rarely covers more than one word. In addition, the morph and syllable models need to back-off quite infrequently (see Table 1), which indicates that higher order n-grams could improve performance. Clustering similar syllables and morphs might improve the language model further. To assure that language models with units of different size could be compared fairly, cross-token acoustic contexts should be implemented in the decoder.

5. Conclusions

Due to the high number of distinct word forms in Finnish, traditional word-based pronunciation lexicons and language models do not work well in Finnish speech recognition. We have applied a data-driven unsupervised MDL-based method to split words into smaller units called morphs. The resulting morphs were able to generate all word forms while attaining reasonable perplexity results. Also the speaker dependent speech recognition experiments were encouraging. The overall word error rate obtained with morphs (32%) was significantly lower than word error rates for syllables (44%) and words (56%).

6. Acknowledgements

We thank the Finnish Federation of the Visually Impaired and the Departments of Phonetics and General Linguistics of the University of Helsinki for providing the speech data. We also thank the Finnish news agency (STT) and the Finnish IT center for science (CSC) for the text data. This research was partly funded by the Finnish National Technology Agency (TEKES) as a part of the USIX project.

7. References

[1] J. Kneissler and D. Klakow, "Speech recognition for huge vocabularies by using optimized sub-word units," in *Proceedings of Eurospeech 2001*, 2001.

[2] W. Byrne, J. Hacič, P. Ircing, F. Jelinek, S. Khudanpur, P. Krbec, and J. Psutka, "On large vocabulary continuous speech recognition of highly inflectional language — Czech," in *Proceedings of Eurospeech 2001*, Aalborg, Denmark, 2001, pp. 487–489.

[3] M. Huckvale and A. Fang, "Using phonologically-constrained morphological analysis in continuous speech recognition," *Computer Speech and Language*, vol. 16, 2002.

[4] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, pp. 21–30, 2002.

[5] J. Rissanen, "Stochastic complexity in statistical inquiry," *World Scientific Series in Computer Science*, vol. 15, 1989.

[6] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," in *Proceedings of Eurospeech 1997*, 1997.

[7] M. Kurimo, "Using self-organizing maps and learning vector quantization for mixture density hidden Markov models," Ph.D. dissertation, Helsinki University of Technology, 1997.

[8] X. L. Aubert, "An overview of decoding techniques for large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 88–114, Jan. 2002.

[9] J. Goodman, "A bit of progress in language modeling," *Computer Speech and Language*, October 2001.