

Multigram-based Grapheme-to-Phoneme Conversion for LVCSR

M. Bisani and H. Ney

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen – University of Technology, D-52056 Aachen, Germany
{bisani,ney}@informatik.rwth-aachen.de

Abstract

Many important speech recognition tasks feature an open, constantly changing vocabulary. (E.g. broadcast news transcription, spoken document retrieval, ...) Recognition of (new) words requires acoustic baseforms for them to be known. Commonly words are transcribed manually, which poses a major burden on vocabulary adaptation and inter-domain portability. In this work we investigate the possibility of applying a data-driven grapheme-to-phoneme converter to obtain the necessary phonetic transcriptions. Experiments were carried out on English and German speech recognition tasks. We study the relation between transcription quality and word error rate and show that manual transcription effort can be reduced significantly by this method with acceptable loss in performance.

1. Introduction

Recently we have implemented a fully data-driven, language independent grapheme-to-phoneme converter [1]. The basic principle is to apply the joint-multigram approach [2] to the alignment problem and use standard language modelling techniques to model transcription probabilities. This method needs nothing but a training sample in form of a pronunciation dictionary (which need not be aligned at the phoneme level), in order to transcribe new words consistent with the given training data.

Whereas our former study evaluated the performance of the method only on a symbolic level, this work focuses on its application in large vocabulary continuous speech recognition. The next section provides a brief review of the method, the remainder of the paper describes experimental setup and re-

sults. The goal of these experiments was to find out whether the grapheme-to-phoneme method could be used to automatically generalize from the manually transcribed words.

2. Grapheme-to-Phoneme Conversion using Joint Multigrams

The fundamental assumption of the joint multigram model is that for each word its orthographic form and its pronunciation are generated by a common sequence of *graphemes*. A Grapheme, or grapheme-phoneme joint multigram, is a pair $q = (\mathbf{g}, \boldsymbol{\varphi}) \in Q \subseteq G^* \times \Phi^*$ of a letter sequence and a phoneme sequence of possibly different length. (G is the alphabet of letters, Φ the inventory of phonemic symbols.) For example, the pronunciation of “speaking” may be regarded as a sequence of five graphemes:

$$\begin{array}{l} \text{“speaking”} \\ \text{[spi:kɪŋ]} \end{array} = \begin{array}{cccccc} \text{s} & \text{p} & \text{ea} & \text{k} & \text{ing} \\ \text{[s]} & \text{[p]} & \text{[i:]} & \text{[k]} & \text{[ɪŋ]} \end{array}$$

The procedure for inducing the set of graphemes for a language from unsegmented training data is described in [1]. The joint probability distribution $p(\boldsymbol{\varphi}, \mathbf{g})$ is modelled using a standard M -gram:

$$p(q_1^L) = \prod_{i=1}^{L+1} p(q_i | q_{i-1}, \dots, q_{i-M+1}) \quad (1)$$

Phonemic transcriptions are obtained from orthographic forms by searching for the most likely grapheme sequence matching the given spelling and projecting it onto the phonemes.

3. Experiments

We conducted recognition experiments on two tasks: An English dictation task and a German spontaneous conversation task. The experiments were designed to simulate a scenario where an existing recognition system is ported to a larger domain. However we decided to use well-known corpora with existing complete pronunciation dictionaries so that we would be able to see how much performance is lost with respect to a (presumably) optimal pronunciation dictionary.

For each test we selected a subset of “known” words. The choice was based on whether the frequency in the language model training corpus was above a given threshold. Then we used their reference transcriptions to train the grapheme-to-phoneme converter. We used the settings which yielded the best results in our previous study [1]: Grapheme size was restricted to one to two letter and phonemes, marginal trimming and a tri-grapheme model were employed.

The resulting grapheme-to-phoneme converter was then used to transcribe the remaining words of the recognition vocabulary. (The transcriptions of the “known” words were copied verbatim.) The speech recognizer was run with this modified pronunciation dictionary.

The acoustic models remained fixed in all experiments. This reflects the situation of a user adding new words to the recognition vocabulary, who does not have the possibility to build acoustic models. Retraining acoustic models using the modified pronunciation dictionary might be beneficial, since it could allow the acoustic models to compensate the systematic errors made by grapheme-to-phoneme conversion. However, we have not completed these experiments, yet.

For the English tests we used the the Darpa North American Business News (NAB) 1994 development test Hub-1 (dev’94). The recognizer uses a 20k vocabulary. All words were converted to lower case, resulting in the usual 26 grapheme symbols. However for spelled letters we used 26 additional symbols. I.e. “us” ([ʌs]) has two grapheme symbols and “U.S.” ([ju:ɛs]) has two different grapheme symbols. The phoneme set consists of 43 symbols.

The German task was the VerbMobil II (VM2) 1999 development test set, featuring spontaneous scheduling and appointment making dialogs. The recognition dictionary has 11k entries and is based on the Bielefeld Lexicon Database VM-II, version 14.0 (LEXDB) [3]. All words were converted to lower case, resulting in 30 grapheme symbols (including 3 umlauts and sz-ligature). Again, an additional set of 29 spelled letter symbols is used. The phoneme set contains 44 symbols.

Quantitative statistics of both speech corpora is given in table 1.

4. Grapheme-to-Phoneme Results

Figure 1 shows the performance of the grapheme-to-phoneme converter on the symbolic level. Performance is measured by the *phoneme error rate* (PER), which is the Levenshtein distance¹ between automatic transcription result and reference pronunciation divided by the number of phonemes in the reference pronunciation. The *string error rate* (SER) is the relative number of words with at least one error. The “known” words are not included, so PER and SER measures the generalization performance of the converter.

More interesting for performance of the speech recognizer are the frequency weighted error rates. In the *frequency weighted phoneme error rate* (fwPER) each word is weighted by its frequency in the LM training corpus. We calculate the fwPER on the complete dictionary including the “known” words, which reflects the effective correctness of the dictionary.

Overall the transcription error rates are lower on the German task because the spelling is generally much closer to the pronunciation than in English. The generalization performance of the grapheme-to-phoneme converter shows a strong dependence on the amount of training data. Yet, it works relatively well even with very modest amounts of training data. For obvious reasons, the frequency weighted error rates are much lower than the unweighted ones.

¹This is the minimum number of insert, delete and substitute operations required to transform one sequence into the other.

Table 1: Statistics of speech corpora used

Corpus	VerbMobil II		NAB	
	Training CD1-41	Test dev'99	Training WSJ0+1	Test dev'94
Duration	61.5h	1.6h	81.4h	0.8h
Silence	13%	11%	27%	19%
# Speakers	857	16	284	20
# Sentences	36,015	1,081	37,571	310
# Words	701,512	14,662	649,624	7,378
Trigram PP.	—	62.0	—	126.6

5. Speech Recognition Results

The speech recognizer uses features derived from 16 Mel-frequency cepstral coefficients with cepstral mean normalization, and linear discriminant analysis on seven consecutive vectors with an output dimension on 33. Acoustic modeling is based on tri-phones with across-word context using 7001 tied state for NAB and 3501 tied states for VM2. The total number of Gaussian densities is 412k for NAB and 383k for VM2. The language model is a tri-gram for both corpora. The recognition system is described in more detail in [4] and [5]. The baseline performance (with the reference dictionary) is 11.5% WER on NAB and 22.5% WER on VM2.

The recognition results with partly automatically derived dictionaries are shown in figure 2. Since acoustic models were used unaltered there is a mismatch condition which accounts for some of the observed degradation. We find that the frequency weighted phoneme error rate is a very good predictor for the word error rate: On both corpora word error rate increases by roughly 1.7 percentage points per one percent in fwPER. It should be noted that due to how this experiment is laid out, the grapheme-to-phoneme performance is very poor. For both languages much better grapheme-to-phoneme converters are available, and also the method used in this work performs a lot better when more training data is used.

The usefulness of using grapheme-to-phoneme conversion in rapid system development for alphabetic languages is shown in figure 3. For German it seems to be sufficient to provide phonetic transcriptions for about 2000 words, which is less than 20% of the vocabulary in this case. We expect that this result applies to other languages with a close relation between spelling and pronunciation. Also for English one can save half the transcription effort if a 10% lack in performance is acceptable.

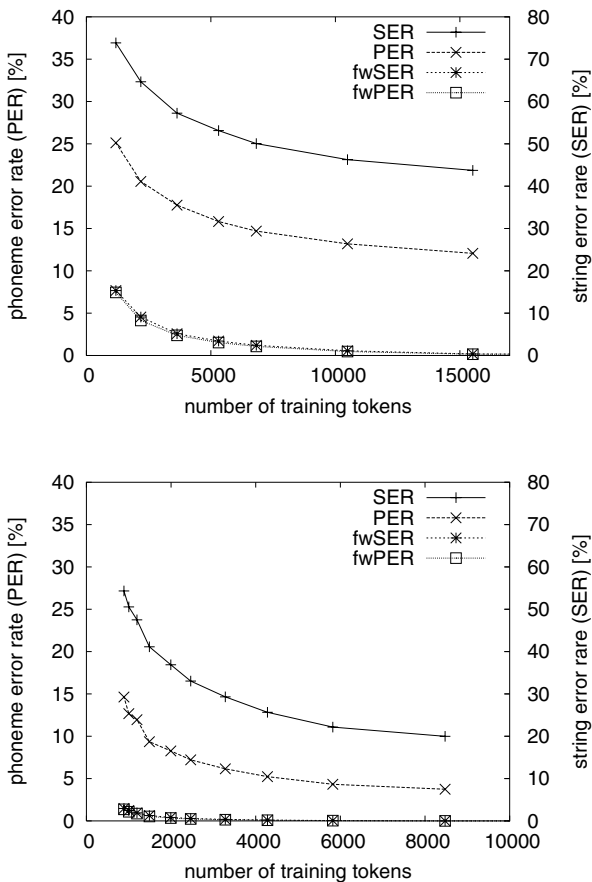


Figure 1: Performance of the grapheme-to-phoneme converter for English (top) and German (bottom): The abscissa is the number “known” words used for training the grapheme-to-phoneme converter. Unweighted PER/SER is measured on unseen words only. Word frequency weighted fwPER/fwSER reflects the overall error. (Note the different scale for phoneme and string error!)

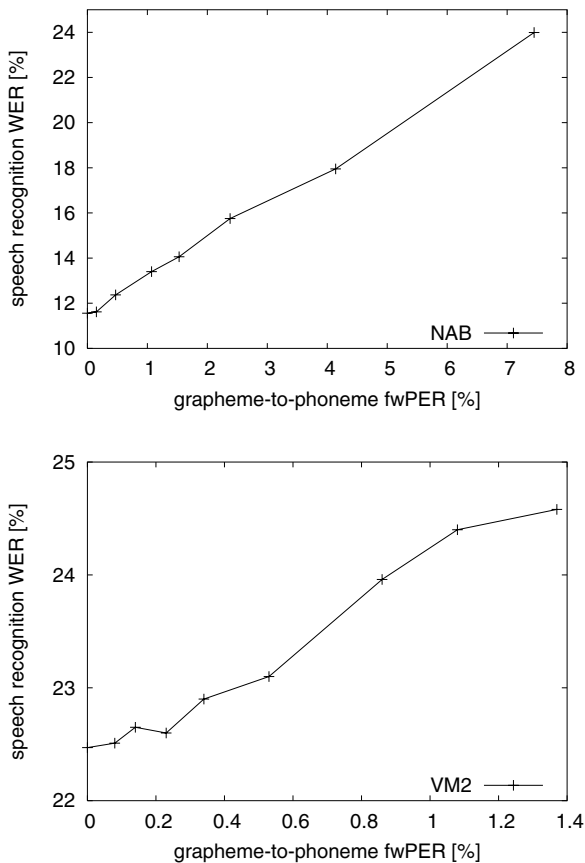


Figure 2: Speech recognition performance on NAB'94 (top) and VerbMobil II (bottom) as a function of grapheme-to-phoneme conversion error rate. The abscissa shows the frequency weighted phoneme error rate of the pronunciation dictionary used.

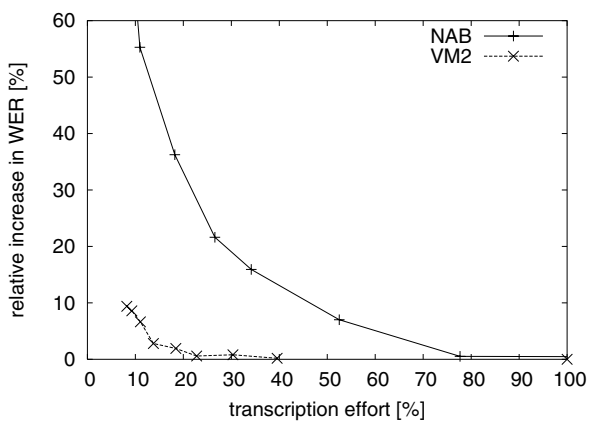


Figure 3: Relative loss in word error rate of the speech recognizer as a function of the percentage of the vocabulary transcribed manually.

6. Summary

We have presented results on the use of grapheme-to-phoneme conversion for building pronunciation dictionaries for large vocabulary speech recognition. Our results show that such dictionaries can be build from very scarce resources. However, the loss in performance with respect to using a fully corrected dictionary, strongly depends on the language.

Acknowledgments: This work was partially funded by the European Commission under the Human Language Technologies project CORETEX (IST-1999-11876).

7. References

- [1] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion," in *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, CO, Sep. 2002, vol. 1, pp. 105 – 108.
- [2] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech*, Madrid, Sep. 1995, pp. 2243 – 2246.
- [3] H. Lungen, K. Ehlebracht, D. Gibbon, and A. P. Q. Simões, "Bielefelder Lexikon und Morphologie in VERBMobil Phase II," Tech. Rep. ISSN 1434-8845, Universitt Bielefeld, November 1998.
- [4] S. Kanthak, A. Sixtus, S. Molau, R. Schlüter, and H. Ney, "Fast search for large vocabulary speech recognition," in *VerbMobil: Foundations of Speech-to-Speech Translation*, W. Wahlster, Ed., pp. 63 –78. Springer, 2000.
- [5] A. Sixtus and H. Ney, "From within-word model search to across-word model search in large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 16, no. 2, pp. 245 – 271, May 2002.