Optimizing phoneme-to-grapheme conversion for out-of-vocabulary words in speech recognition

Bart Decadt, Walter Daelemans*
CNTS Language Technology Group, University of Antwerp
e-mail: {decadt,daelem}@uia.ua.ac.be

Deliverable WP2, September 30, 2001

Abstract

In this report, we present the results of further research on phoneme-to-grapheme (P2G) conversion for Out-Of-Vocabulary items (OOVs), recognized using phoneme recognition, in large vocabulary speech recognition. First, we summarize the results of previous research, and then we start with reporting on several optimization strategies for the Machine Learning technique we used to carry out P2G conversion, and we investigate spelling correction as a post-processing step. Next, as some further error analysis, we compare the Machine Learning technique to a statistical method for P2G conversion. Finally, we take a look at how our P2G converter interacts with the confidence measures used in ESAT's speech recognizer.

1 Introduction

In a large vocabulary continuous speech recognition system, the occurrence of Out-Of-Vocabulary items (OOVs) in the input speech is problematic: as the word is not contained in the lexicon, the speech recognizer can never propose a good solution for that item in the input speech, resulting in bad transcriptions. However, when enhancing the speech recognizer with confidence measures, it becomes possible to more or less reliably identify part of the input as an OOV. The identified items can then be replaced with a phoneme string, and this phoneme string can be converted to graphemes using P2G conversion based on Machine Learning techniques.

We chose Memory Based Learning (MBL) as the Machine Learning technique for the P2G converter (the implementation of MBL we used is called TIMBL (Daelemans et al., 2000)): our hypothesis is that MBL can adapt to the peculiarities of the errors made by the phoneme recognizer, and can provide the necessary robustness and accuracy to the P2G task when provided with sufficient training data (we will investigate this hypothesis in section 4). For more information on TIMBL we refer to (Daelemans et al., 2000).

In the next sections, we summarize the results of our previous research on P2G conversion (described in (Decadt and Daelemans, 2001)), after which we give an overview of the follow-up research we carried out.

*Research funded by IWT in the STWW programme, project ATraNoS (Automatic Transcription and Normalisation of Speech). The research consortium consists of CCL and ESAT (K.U. Leuven), CNTS (U. Antwerpen), and ELIS (R.U. Gent). The project aims at generic basic research on speech recognition and normalization of unrestricted speech with automatic subtitling from speech as a case study. We would like to thank the project partners of ESAT for their cooperation in the work reported here. We would also like to thank our colleagues Véronique Hoste and Erik Tjong Kim Sang for their help.

1.1 Previous research and results

Our previous research was concerned with (i) investigating the feasibility of P2G conversion using MBL, by doing experiments with a dataset to which different levels of artificially generated noise was added, (ii) carrying out experiments with phoneme recognizer data, and (iii) providing a detailed error analysis.

To investigate the feasibility of P2G conversion, we did experiments with CELEX (Baayen, Piepenbrock, and van Rijn, 1993), a lexicon containing 173k Dutch words and their pronunciation, to which artificial noise was added. The addition of noise to CELEX was done such that it simulated the type of confusion between sounds a phoneme recognizer may make: a certain amount of phonemes in the lexicon was replaced with one of its three *closest* phonemes.

We conducted *ten-fold cross-validation* (10CV) experiments at various levels of noise (ranging from 0% to 50% noise, in steps of 5%), and with different parameter settings for TIMBL: we experimented with IB1-IG, the standard MBL algorithm, with k, or the number of *nearest neighbors* to extrapolate from, set at 1, 3 or 5, and with IGTREE, a decision tree heuristic approximation of MBL. For each percentage of noise, we found that IB1-IG with k set at 1 produces the best conversion accuracy at word and at grapheme level: when the phoneme strings are error-free, P2G is almost an easy task (91.4% word level accuracy with 99.1% grapheme level accuracy). However, performance decreases steadily while the amount of noise increases: with 25% to 30% of artificial noise added (the typical error rate of a phoneme recognizer), P2G conversion is done with 53.0% to 47.7% word level and 93.1% to 92.0% grapheme level accuracy.

The results from the experiments with real data, produced by ESAT's phoneme recognizer (for more information, see: (Demuynck et al., 1998), (Duchateau, 1998), (Demuynck et al., 2000) and (Demuynck, 2001)) with a recording from Corpus Gesproken Nederlands (*Spoken Dutch Corpus* [1]), were quite different: in our artificial noise generation method we only simulated substitutions and did not take into account some other typical phoneme recognizer errors, i.e. insertions and deletions. The best scoring TIMBL parameter setting in these experiments was IB1-IG with k set at 5: the accuracy at word level (46.5%) was more or less in line with the previous experiments, but the accuracy at grapheme level (77.4%) was much lower. We also computed accuracy percentages for the OOVs in the dataset: 63.3% accuracy at grapheme level, but only 6.9% accuracy at word level.

The low accuracy at word level can be explained by the high percentage of phoneme strings with phoneme deletions in the OOVs of the dataset (44.7%): because we need a one-to-one correspondence between phonemes and graphemes for doing P2G conversion with TIMBL, we are not able to convert a phoneme string with deletions to a completely correct word. Though only few OOVs are correctly predicted, the output of TIMBL should still be more or less readable: 17.1% of the OOVs was predicted with at most 1 error, and 32.2% with at most two errors.

Finally, we conducted a detailed error analysis on TIMBL's output. When we analyzed the output of the experiments with an error-free CELEX lexicon, we found that most errors are due to ambiguous phonemes (because of spelling conventions, assimilation processes or morphological or syntactic rules) and loan words with a spelling which is atypical for Dutch. In the output of the experiments with the dataset from ESAT's phoneme recognizer, we noticed that *compound graphemes* (introduced to achieve a one-to-one correspondence between phonemes and graphemes) are more difficult to predict than *normal* graphemes.

[1] Sponsored by the Dutch NWO and the Flemish IWT, see `http://lands.let.kun.nl/cgn/ehome.htm`

1.2 Further research

As further research on P2G for OOVs, we first of all tried to optimize the P2G converter. In order to minimize the amount of errors due to phoneme deletions, ESAT tuned its phoneme recognizer to produce less phoneme deletions and created a new dataset. The results of experiments with this dataset are reported on in section 2.1. A second strategy for optimizing the P2G converter, is trying to train TIMBL more on the peculiarities of OOVs by increasing the amount of OOVs in the dataset. This will be investigated in section 2.2. A final optimization strategy is to apply an optimization algorithm to the MBL technique used, which does feature selection and searches for an optimal parameter setting. The results of this algorithm are given in section 2.3.

Next, we investigate spelling correction as a possible post-processing step. A lot of words in TIMBL's output contain only one or two errors: spelling correction could be a means to correct these errors. The results of adding a spelling corrector to the system are reported on in section 3.

In section 4, we compare the MBL technique to a purely probabilistic approach to P2G conversion. As such, we can test the hypothesis that MBL can adapt to the peculiarities of the errors made by the phoneme recognizer.

Finally, in section 5 we take a brief look at how the P2G converter interacts with the confidence measures which will be used in ESAT's large vocabulary speech recognizer.

2 Optimization of the P2G-converter

2.1 Memory Based Learning on a dataset containing less deletions

The high number of words with phoneme deletions in the OOVs of ESAT's dataset (44.7%) was a source of many errors at word level for these OOVs. Reducing the number of phoneme deletions could result in a higher word level accuracy. The dataset we used for our previous experiments had a phoneme recognition error rate of ~25.6%.

ESAT created a new dataset with 20% less deletion errors, but 60% more insertions and 15% more substitutions. The overall phoneme recognition error rate of this dataset is a bit higher (~29%). However, this higher error rate should not make the task much more difficult, as the current architecture of our P2G converter can handle substitutions and insertions (for insertions, an *empty grapheme* should be predicted). In this dataset, the percentage of OOVs containing phoneme deletions has now dropped to 37.6% (compared to 44.7%).

With these new data, we ran 10CV experiments. These experiments were done in the same way as the previous ones (described in (Decadt and Daelemans, 2001)), except that we did no longer investigate whether including spelling context (of the previous/next word) in the feature set of a phoneme leads to higher accuracy as our previous experiments suggested that this is not the case.

The results, at grapheme and word level, and for the complete dataset and only the OOVs, are presented in Table 1. The best scoring TIMBL parameter setting, most clearly for the OOVs, is IB1-IG with k set at 5.

The conversion accuracy on the complete dataset is a bit worse: 75.7% compared to 77.3% at grapheme level, and 43.9% compared to 46.5%. However, for the OOVs only, the accuracy slightly increases with 0.7% at word level (7.6% compared to 6.9%; a 10.1% gain) and 0.5% at grapheme level (63.8% compared to 63.3%; a 1.6% gain).

The number of words converted with only one or two errors remains more or less the same: 18.4% of the OOVs contain at most one error, and 34.2% at most two errors.

		IB1-G			IGTREE
		$k=1$	$k=3$	$k=5$	
COMPLETE DATASET	graph. level acc.	74.2	75.7	**75.7**	74.4
	word level acc.	43.9	**44.1**	**43.9**	43.8
OOVS IN DATASET	graph. level acc.	59.5	63.2	**63.8**	60.5
	word level acc.	6.1	7.1	**7.6**	6.1

Table 1: The results (acc. in %) of experiments with a dataset containing less deletions

		IB1-G			IGTREE
		$k=1$	$k=3$	$k=5$	
COMPLETE DATASET	graph. level acc.	76.0	**76.5**	75.7	76.1
	word level acc.	**46.12**	**42.5**	38.6	46.1
OOVS IN DATASET	graph. level acc.	59.9	**63.7**	**63.9**	60.8
	word level acc.	6.2	**7.0**	5.3	6.3

Table 2: The results (acc. in %) of experiments with a dataset in which each OOV was doubled

2.2 Memory Based Learning on a dataset containing more OOV items

In the dataset used for training the P2G, the number of OOVs is rather small: there are only 9k OOVs, in contrast to 120k non-OOVs. Due to their infrequency in the dataset, our P2G converter is not very well trained on the peculiarities of these OOVs (assuming that there are any).

A straightforward solution for this problem is to create a new dataset, in which each OOV is doubled - which results in 18k OOVs [2]. With such a dataset we ran some experiments, the results of which are presented in Table 2.

The best parameter setting is IB1-IG with k set at 3: however, we do not get a striking increase in accuracy; only 0.1% at word-level and 0.4% at grapheme level for the OOVs in the dataset.

We could, of course, increase the presence of OOVs in the dataset even more, simply by tripling, quadrupling, ... these items, but it is not very likely that this will lead to better results: there probably is not a lot of regularity in the OOVs - they are mainly compounds, loan words (mainly from French, German and English) and proper names. The latter two categories are known to have unsystematic spelling-pronunciation correspondences [3].

2.3 Applying an optimization algorithm to Memory Based Learning

To optimize TIMBL's parameter settings, we used the algorithm described in Figure 1 [4]. We ran the algorithm with ESAT's first dataset, starting with a feature list containing 11 features: the phoneme to be converted and the five preceding and following phonemes.

[2]Suggested by dr. ir. J. Duchateau (ESAT).

[3]The dataset does not contain the necessary tags to give accuracy figures distinguishing compounds, loan words and proper names among the OOVs.

[4]This algorithm was already implemented by our colleagues V. Hoste and A. Kool for optimizing TIMBL on the SENSEVAL task. We made same minor changes to the implementation to adapt it to our task, P2G conversion.

1. Determine and save the default score: run TIMBL with IB1-IG and k set at 1, and use all features in the feature list.
2. Search for the best feature combination:
   (a) set *Feature-to-Remove* to empty, and for each feature f in the feature list, do:
      - run TIMBL with IB1-IG and k set at 1, and do not use feature f
      - if the current score is higher than the default score, then set the default score to the current score, and set *Feature-to-Remove* to feature f

   (b) if *Feature-to-Remove* is empty, then go to the next step;
   else do:
      - permanently remove feature stored in *Feature-to-Remove* from the feature list,
      - reset *Feature-to-Remove* to empty
      - go back to step 2.(a)
3. Search for the best value for the parameters *weighting* ($w = \{$*gain ratio, info-gain, chi-squared, shared variance*$\}$) and *nearest neighbors* ($k = \{1, 3, 5, 7, 9, 11, 13, 15\}$):
   - set *Best-Weighting-Value* to empty, and for each value w_i in *weighting*, do :
     - set the default score to zero, set *Best-Nearest-Neighbors-Value* to empty, and for each value k_j in *nearest neighbors*, do:
       * run TIMBL with IB1-IG, *weighting* set at w_i and *nearest neighbors* at k_j
       * if the current score is higher than the default score, then set *Best-Weighting-Value* to w_i and *Best-Nearest-Neighbors-Value* to k_j
4. Return the features that stayed in the feature list, and the values stored in *Best-Weighting-Value* and in *Best-Nearest-Neighbors-Value.*

Figure 1: The optimization algorithm for TIMBL

The algorithm started with a score of 76.2% at grapheme level for default settings (we did not take into account the score at word level). When the algorithm was finished, we received the following output:

- the **feature list** was reduced to the phoneme to be converted and the two preceding and following phonemes;
- the best setting for **weighting** is *info-gain*;
- the best setting for **nearest neighbors** is 3.

Running TIMBL with these settings results in a score at grapheme level of 76.8%. This is a gain in grapheme level accuracy of 0.6%. Though obviously some optimization occurred, this result is not higher than the highest result we achieved in our first experiments with this dataset, i.e. an accuracy of 77.4% at grapheme level.

The problem with the algorithm in Figure 1 is that it searches for an optimal feature combination separately from the search for the optimal setting for the parameters *weighting* and *nearest neighbors*. Ideally, for each feature it tries to remove from the feature list, the algorithm should search which parameter setting scores best and use that score as the new default score. Though not too difficult to implement, this version of the optimization algorithm is not computationally feasible.

3 Spelling correction as a post-processing step

In the error analysis of our previous experiments (see (Decadt and Daelemans, 2001), section 3.3), we noted that (i) short words, on average, tend to have more errors than expected, and (ii) words containing 4, 5 and even 6 errors were quite frequent in TIMBL's conversions for the OOVs. On the basis of these observations, we did not expect an

<table>
<tr><th colspan="2">Correctly converted words: 612 (6.9%)</th><th colspan="2">Incorrectly converted words: 8280 (93.1%)</th><th></th></tr>
<tr><td>marked as correct</td><td>487</td><td>marked as correct</td><td>1291</td><td></td></tr>
<tr><td>marked as incorrect with suggestions</td><td>109</td><td>marked as incorrect with suggestions</td><td>3018</td><td></td></tr>
<tr><td>marked as incorrect without suggestions</td><td>16</td><td>marked as incorrect without suggestions</td><td>3971</td><td></td></tr>
<tr><th colspan="2">Loss in accuracy (correct words marked as incorrect)</th><th colspan="2">Gain in accuracy (incorrect words with correct suggestion)</th><th>Total accuracy</th></tr>
<tr><td colspan="2" rowspan="3">-1.4%</td><td>considering only the 1st suggestion:</td><td>+2.4%</td><td>7.8%</td></tr>
<tr><td>considering the first 3 suggestions:</td><td>+4.1%</td><td>9.6%</td></tr>
<tr><td>considering all suggestions:</td><td>+4.8%</td><td>10.3%</td></tr>
</table>

Table 3: The result of spelling correction on the output of TIMBL with IB1-IG and k set at 5

enormous improvement from using a spelling corrector for post-processing. Assuming that only words with 1 or 2 errors have a reasonable chance of being corrected by a spelling corrector, the maximum increase in word level accuracy we may expect, is ~25%, which still is quite an improvement compared to the ±7% word level accuracy we have now.

For our experiments, we did not develop a spelling corrector specifically adapted to our task: we used *iSpell*, UNIX' spelling corrector. The Dutch lexicon for *iSpell* contains 114k words, and a list of affixes to form new words with. *iSpell* can be used in an automatic mode in which each input word is checked for correctness: each word receives a mark (correct or incorrect), and for the incorrect ones, *iSpell* gives a list of suggestions, if there are any.

In Table 3, we present the results of running *iSpell* on the list of P2G conversions for the OOVs in ESAT's dataset (obtained by running TIMBL with IB1-IG and k set at 5 on the dataset with the least errors). These results are rather indicative: we could get better results if we use a spelling corrector specifically adapted to our task, with e.g. a lexicon containing more proper names.

It is clear that *iSpell* is not able to correct most of the words containing only one or two errors: the gain in word level accuracy is 2.4% (taking into account only the 1^{st} suggestion) to 4.8% (considering all suggestions). Moreover, spelling correction also degrades performance: we lose 1.4% word level accuracy because of correctly converted words marked as incorrect by *iSpell*. This does not mean that spelling correction is useless as a post-processing step: gaining 4.8% in word level accuracy (resulting in 10.3%) means an improvement of ~70% compared to our previous result, 6.9%. Furthermore, *iSpell* is able to mark 84.4% of the incorrectly converted words as incorrect. This information can be used in the final transcriptions of the speech recognizer enhanced with our P2G converter, e.g. it can be presented with color codes.

4 Further error analysis

In section 1, and in (Decadt and Daelemans, 2001), we have put forward the hypothesis that TIMBL, or rather MBL, is able to adapt to the peculiarities of the errors made by the

		STATISTICAL APPROACH	TIMBL IB1-IG, $k = 5$
COMPLETE DATASET	graph. level acc.	70.5%	77.4%
	word level acc.	30.0%	46.5%
OOVS IN DATASET	graph. level acc.	60.2%	63.3%
	word level acc.	3.0%	6.9%

Table 4: The results (acc. in %) of a statistical approach to P2G conversion

phoneme recognizer. One way to give evidence in favor of this hypothesis, is to compare TIMBL's performance on the P2G conversion task to a statistical method.

The statistical method is a very basic approach to P2G conversion: we simply convert a phoneme to the most frequent grapheme for that phoneme, e.g.: if the phoneme /p/ corresponds in 78% of the cases with the grapheme *p* and in only 22% with *b*, then we always convert /p/ to *p*.

In the statistical approach, adaptation to the peculiarities of the phoneme recognizer errors is not possible, because context is not taken into account and each phoneme has only one possible grapheme. TIMBL, on the other hand, can take the previous and following phonemes into account, and can give multiple graphemes for a particular phoneme: if TIMBL adapts to the errors, it should score better. The results of the statistical approach, compared to the best results obtained with TIMBL are listed in Table 4.

For the complete dataset as for the OOVs only, at grapheme and at word level, TIMBL scores better than the statistical method, though the difference in performance is more outspoken in the results for the complete dataset. This indicates that there are probably few regularities for P2G conversion in the collection of OOVs.

5 Interaction with ESAT's confidence measures

Finally, to test the interaction of our P2G converter with the confidence measures, used by ESAT's speech recognizer, for identifying possible OOVs in the input speech, we ran an experiment in which we used the dataset with the least errors as training material, and a separate dataset containing 3.6k words as test material [5]. This test-set contained some additional tags: one tag for identifying the OOVs (3.5%) and another for identifying speech recognizer errors (14.7%). We found an overall accuracy at word level of 55.2%: more specifically, we obtained a 7.9% word level accuracy for the OOVs, 19.2% for the recognition errors, and 59.9% for the correct words.

As ESAT's research on confidence measures is not yet finished, and as they are not yet added to the speech recognizer, we have to estimate how our P2G converter will interact with the speech recognizer on the basis of figures found in the literature on confidence measures. From (Kemp and Schaaf, 1997) we know that, for a speech recognition task with a word error rate (WER) as the one found on the separate test-set (14.7%), the threshold in the confidence measure can be adjusted so that about 75% of the recognition errors are tagged as *uncertain words*, and thus missing 25% of the errors, while tagging wrongly only 10% of the correctly recognized words.

Making the assumption that (i) the 75% correctly labeled *uncertain words* are converted with the OOV word level accuracy of 7.9%, and (ii) the 10% incorrectly labeled *uncertain words* are converted with the 59.9% accuracy for correct words, then the WER

[5]This experiment and its result are also reported on in (Decadt et al., 2001). The figures on how well our P2G converter interacts with ESAT's speech recognizer were calculated by dr. ir. J. Duchateau (ESAT)

of the speech recognizer, enhanced with our P2G converter, on the separate test-set would increase to 16.0% (instead of 14.7%).

This does not mean that the resulting transcriptions are less readable: though only 19.2% of the 14.7% speech recognition errors in the test set are correctly converted, 41.0% is converted with at most 1 error, and 62.6% with at most 2 errors. A lot of these words can be easily interpreted by a reader. Below are some examples of words wrongly recognized by ESAT's speech recognizer, and TIMBL's conversions of the phoneme strings for these words.

CORRECT WORD	RECOGN. ERROR	CONVERSION & PHON. STRING
programmaproducent (*program producer*)	programma producent	programaprodusent /prOGrAmAprOdYsEnt/
gespreksonderwerp (*topic of conversation*)	gesprek zonder werk	gespreksonberwerp /G@spreksOnd@r@wEr@/
speelgoedmitrailleur (*toy machine gun*)	speelgoed moet hier	spergoetnietrijer /sperGutnitrKj-yr/

For the recognition errors, only the first word is readable (the speech recognizer did not recognize it as a compound but as two separate words). The other two words are nonsense: "gesprek zonder werk" means *conversation without work* and "speelgoed moet hier" means *toys must here*. On the other hand, the first two conversions made by TIMBL are closer to the correct word and more readable: "programaprodusent" contains two errors and "gespreksonberwerp" only one. The last conversion, "spergoetnietrijer", containing 9 errors, is an example of a loan word with a spelling atypical for Dutch. Due to the fact that both the phoneme recognizer and the P2G converter are trained to produce strings typical for Dutch, these words will always result in bad conversions.

Finally, spelling correction again proves to be useful as post-processing: it increases the word level accuracy for OOVs to 8.7%, for recognition errors to 20.9%, and to 60.1% on average over all words in the test-set. The word error rate of the speech recognizer combined with our P2G converter then drops from 16.0% to 15.4%. From the examples above, only the second word ("gespreksonderwerp") could be corrected by *iSpell*.

6 Conclusion and further research

We found two ways in which to improve the accuracy of the phoneme-to-grapheme conversion process: tuning the phoneme recognizer to produce fewer deletions, and using spelling correction as a post-processing step. Although word level accuracy on the OOVs is not very high, we showed that TIMBL did learn to adapt to the errors of the phoneme recognizer to a certain extent. Even when in an integration with the speech recognizer the total WER increases, readability of the output can nevertheless be improved with this method.

We believe the spelling error correction post-processing can be made more reliable by using lexicons and correction strategies tuned to OOVs and tuned to the type of errors the P2G module makes. More work can also be done on optimization of feature selection and algorithm parameters for the learning task, and the approach should be further tested in combination with different types of confidence measures.

7 References

Baayen, R. H., R. Piepenbrock, and H. van Rijn. 1993. *The* CELEX *lexical data base on* CD-ROM. Linguistic Data Consortium, Philadelphia, PA.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch. 2000. TiMBL: Tilburg memory based learner, version 3.0, reference guide. ILK Technical Report 00-01, Tilburg University. Available from `http://ilk.kub.nl`.

Decadt, B. and W. Daelemans. 2001. Phoneme-to-grapheme conversion for out-of-vocabulary words in speech recognition. Technical report, ATraNoS Project (IWT-STWW), CNTS Language Technology Group, University of Antwerp.

Decadt, B., W. Daelemans, J. Duchateau, and P. Wambacq. 2001. Phoneme-to-grapheme conversion for out-of-vocabulary words in large vocabulary speech recognition. In *Proceedings of the ASRU Workshop*, Madonna di Campiglio, Italy, December. IEEE. To appear.

Demuynck, K. 2001. *Extracting, modelling and combining information in speech recognition*. Ph.D. thesis, K.U.Leuven, ESAT, February.

Demuynck, K., J. Duchateau, D. Van Compernolle, and P. Wambacq. 1998. Fast and accurate acoustic modelling with semi-continuous HMM. *Speech Communication*, 24(1):5–17, April.

Demuynck, K., J. Duchateau, D. Van Compernolle, and P. Wambacq. 2000. An efficient search space representation for large vocabulary continuous speech recognition. *Speech Communication*, 30(1):37–53, January.

Duchateau, J. 1998. *HMM based acoustic modelling in large vocabulary speech recognition*. Ph.D. thesis, K.U.Leuven, ESAT, November.

Kemp, T. and T. Schaaf. 1997. Estimating confidence using word lattices. In *Proceedings of EuroSpeech vol. II*, pages 827–830, Rhodes, Greece, September.