Phonemisation

Benedikt Adelmann and Tim Dobert

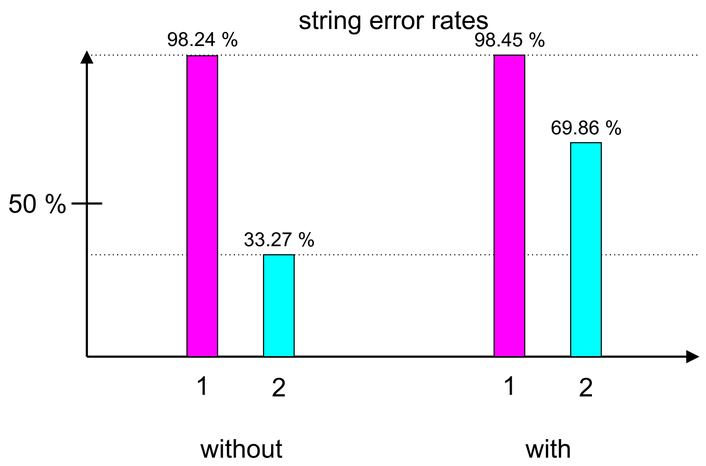
Lab Group, Vertiefungsmodul Sprachverarbeitung bei Dr. Timo Baumann, Sommersemester 2016

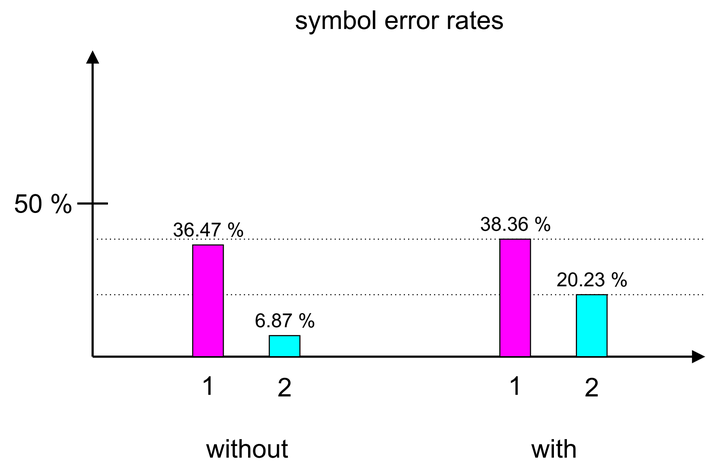
Our task was to analyze grapheme-to-phoneme conversion (in short: G2P) performance without and with morpheme boundaries for German and to study the effect of learned boundaries on G2P. For this purpose, we used a tool called Morfessor that promises to reconstruct morpheme boundaries from lists of words of a given language and a tool called Sequitur G2P, 'a data-driven grapheme-to-phoneme converter [that] can be applied to any monotonous sequence translation problem, provided the source and target alphabets are small (less than 255 symbols), [and] has no no built-in linguistic knowledge' (Sequitur README).

Unfortunately, our lab group dissolved over the course of the semester so that we were unable to complete all the tasks we had planned. Nevertheless, the results we did obtain are summarized here.

An illustration of our experimental setup can be found in the middle of this poster.

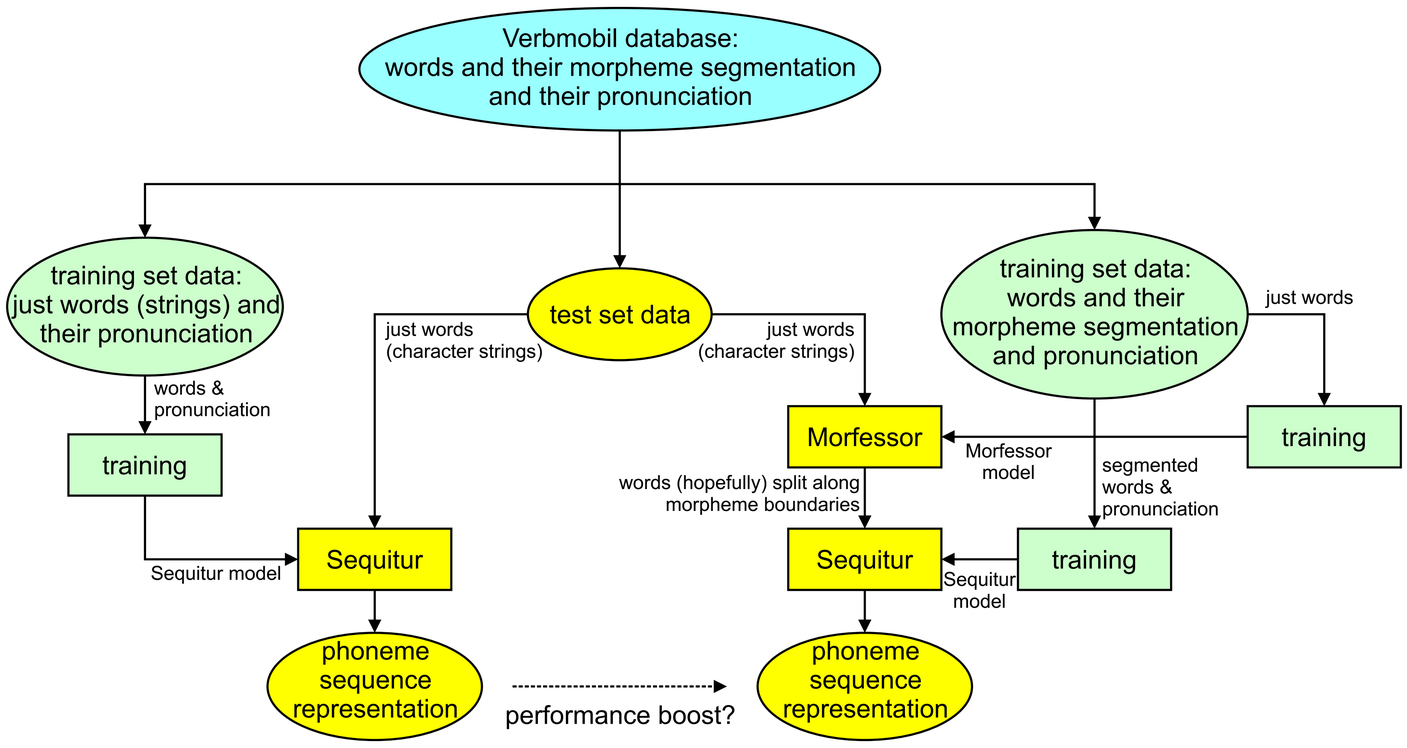
We trained a total of four Sequitur models, two with morpheme boundaries and two without morpheme boundaries.
The models without morpheme boundaries are classical G2P models. They were trained from a list of words and their correct phonetic transcriptions. Both the words and the correct transcriptions were taken from the Verbmobil database for German. Some of the words in the database offered several possible phonetic transcriptions. Such words were duplicated before training, one for each possible transcription.
The models with morpheme boundaries were trained using a list of words split up at morpheme boundaries – instead of just words. The split words were taken from the Verbmobil corpus as well, which contains one or more morphological segmentations for each word. If there were multiple possibilities, words were duplicated again. The phonetic transcriptions used for training remained the same.
For each type of model we trained two models of different order, one of order 1 (unigrams) and one of order 2 (bigrams).
The classical models were tested using test data that comprised roughly 10 % of the Verbmobil database.
The models with morpheme segmentation were tested by having Morfessor segment the test data input and forwarding Morfessor's output to Sequitur.

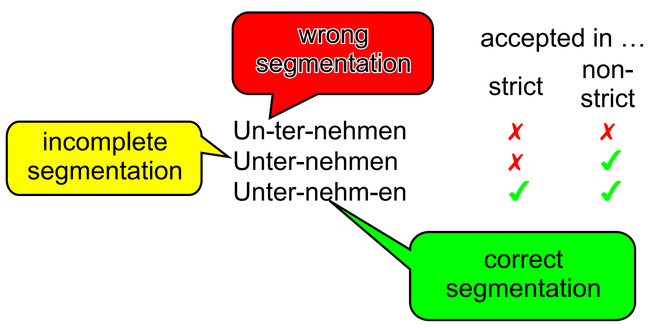




The results, which are shown in the table on the right and visualized in the above diagrams, show that taking morpheme boundaries into account does not only lack any positive effect on the G2P quality but even impairs the overall G2P performance, both in terms of string error rate and symbol error rate: For both unigram and bigram models the error rates were higher for the models with morpheme boundaries than for those without, although this is only significant for the bigram models. As one would expect, bigram models performed overall notably better than the according unigram models.
One possible reason for the deterioration is the considerably bad performance of the Morfessor morpheme segmentation – see below. This also explains the greater impact on the bigram models which heavily rely on input symbol sequences – a bad morpheme segmentation will lead to missing morpheme boundary symbols, resulting in devastating changes in the input symbol sequence. Higher-order models could be able to compensate for that – but the results question the usefulness of morpheme segmentation itself as a preprocessing step for G2P, at least if the segmenting model is too imprecise.

		errors	
boundaries	order	string	symbol
without	1	98.24 %	36.47 %
without	2	33.27 %	6.87 %
with	1	98.45 %	38.36 %
with	2	69.86 %	20.23 %





Morfessor Evaluation

Although it was not our primary assignment, we also evaluated the performance of the model Morfessor delivered. This is reasonable as we cannot expect G2P to benefit from a poor morpheme segmentation – quite the contrary, a bad morpheme segmentation can be considered likely to *impair* subsequent G2P attempts.
First we split the Verbmobil database randomly into a training set that comprised roughly 90 % of the words (more precisely, each word was put into the test set with a probability of 10 % and into the training set otherwise). The correctness of the segmentation Morfessor delivered for the test-set words was assessed with respect to two different correctness measures: a 'strict' one and a 'non-strict' one.
The 'strict' correctness measure considered a segmentation correct if and only if the morpheme boundaries were set at the exactly same positions as in the Verbmobil database. The 'non-strict' correctness measure also considered a segmentation correct if morpheme boundaries were missing. The figure on the left shows example segmentations of 'Unternehmen' accepted or not accepted by the different measures, where the Verbmobil segmentation was 'Unter-neh-men'.
As Morfessor warned us that our test set was 'too small for our sample size', we repeated training and test (in the same manner as described above) with both training and test set comprising roughly 50 % of the Verbmobil database (randomly assigned again).

The results the evaluation yielded are shown in the table on the left and visualized in the diagrams below. They can be summarized as follows: Morfessor found correct morpheme boundaries in about 80 % of the cases, but hardly ever found all of them (just 11.7 % to 14.9 % of the cases). Interestingly, with strict correctness measure, *f*-score, precision and recall were *higher* for the 50 % model (where the training set was smaller and the test set larger) while the corresponding values only dropped marginally when evaluation used the non-strict correctness measure.
As the G2P model would be trained from maximally segmented words (as they can be found in the Verbmobil database), we expected this to be a problem. More precisely, the input data for the G2P training would be words split at all possible morpheme boundaries, so the G2P model should be expected to work best for (test) input words that are also split at all possible morpheme boundaries. If the Morfessor preprocessing does not give all those possible segmentations points, the task of recognizing the input appropriately should become more difficult, resulting in a G2P performance worse than without morpheme boundaries.

Specifically, Timo Baumann's example word for which G2P should be improved by morpheme segmentation, 'Comicheldin', was split into 'Co-mich-el-di-n' by Morfessor (correct: 'Comic-held-in').

test data	strict	*f*-score	precision	recall
10 %	no	0.897	81.4 %	100.00 %
10 %	yes	0.101	11.7 %	8.82 %
50 %	no	0.885	79.4 %	100.00 %
50 %	yes	0.12	14.9 %	10.00 %