Proceedings of Eurospeech, Berlin, 1459-1466, 1993

TabTalk: REUSABILITY IN DATA-ORIENTED GRAPHEME-TO-PHONEME CONVERSION

Walter Daelemans and Antal van den Bosch

ITK Tilburg University
P.O. Box 90153, NL-5000 LE Tilburg
walter@kub.nl, antalb@kub.nl

In: Proceedings of Eurospeech 1993, Berlin.

ABSTRACT

In the traditional (knowledge-based) approach to the design of grapheme-to-phoneme modules in text-to-speech systems, it is claimed that various explicitly coded, language-specific, linguistic knowledge sources are necessary for a good performance. Due to knowledge acquisition bottlenecks, this implies long development cycles. As an alternative, we propose to use inductive methods from machine learning in a simple combined Trie Search and Similarity-Based Reasoning approach and show that, for Dutch, its performance is better than that of the knowledge-based approach and backpropagation learning. Furthermore, we show that our approach is reusable for any language for which a training corpus exists.

Keywords: grapheme-to-phoneme conversion, text-to-speech, trie search, similarity-based reasoning, machine learning

INTRODUCTION

The larger part of research on grapheme-to-phoneme conversion focuses on developing systems that implement various levels of language-specific linguistic knowledge. It is generally assumed that this is essential to solving the task. A clearly disadvantageous consequence of this strategy is the fact that numerous knowledge acquisition bottlenecks have to be passed during development. Furthermore, language-specificity of a grapheme-to-phoneme model tends to be incompatible with reusability of the developed implementation, i.e., for each language, a specific set of rules and principles has to be found in order to successfully run the model. MITalk [1] is a classic example of such a model for English; for Dutch, Morpa-cum-Morphon [6] can be considered state-of-the-art.

In this paper, we present a model the construction of which is simple and does not involve linguistic engineering, nor the inclusion of language-specific knowledge, viz. a combination of Trie Search and Similarity-Based Reasoning. These simple data-oriented machine learning techniques are applied to corpora of word-pronunciation pairs, of which the existence is the only prerequisite for applying the model. After a description of the two machine learning techniques, we present performance results of our model for English and for Dutch. For the latter language, we compare our results with those of Morpa-cum-Morphon [6] on the same test material. Secondly, we present a comparison of the performance of our model on corpora of English, Dutch and French word-pronunciation pairs.

TabTalk

Van den Bosch & Daelemans [2] present two machine learning techniques, Instance-Based Learning [1] and Table Lookup with defaults which they train on grapheme-to-phoneme conversion. Both techniques take as their basis a large corpus of word-pronunciation pairs, store (parts of) this corpus in a memory base and apply a certain retrieval mechanism in order to categorise unseen test cases as correctly as possible. However, there are some essential differences between the two approaches.

Table Lookup can be seen as optimized, generalised lexical lookup. This approach has as its major disadvantage the fact that it only works for words that are stored in the lexicon and not for new words. The Table Lookup model solves this problem of lacking generalisation power and efficiency by *compressing* it into a grapheme-to-phoneme *lookup table*. The main strategy behind this compression is to dynamically determine which left and right contexts must be minimally known to be able to map a single grapheme to its corresponding phoneme with absolute certainty (in the training corpus). Generalisation is achieved because of the fact that unknown words contain known substrings of graphemes, and by adding a default table that predicts the most probable transcription given an unknown string of graphemes.

In our present implementation of TabTalk, compression of a training corpus of word-pronunciation pairs is taken one step further by compressing the lookup table into a *trie*. Finding a phonemic mapping of a grapheme is done by a search through the trie. An example of such a search path is shown in Figure 1, in which the pronunciation of the $< a >$ in $< behave >$ is retrieved.

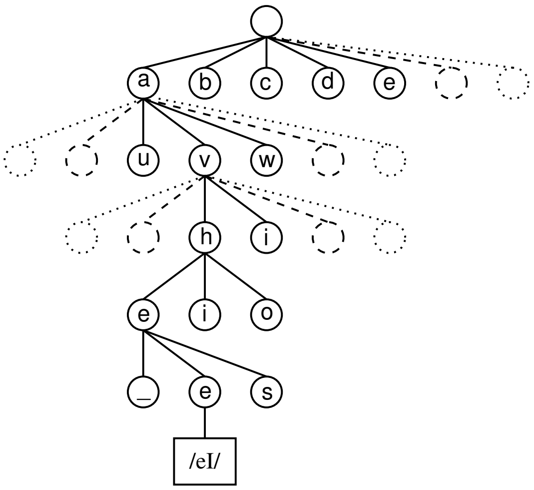

Figure 1. *Retrieval of the pronunciation of* $< a >$ *in* $< behave >$ *by trie search.*

First, the node denoting the focus grapheme $< a >$ is accessed from the top node. The nodes in the layer below the $< a >$-node are all graphemes which occurred immediately right adjacent to $< a >$ in the training corpus. At this point, the pronunciation of $< a >$ is still ambiguous, although the trie might contain information about what would be the most probable pronunciation. With the extension $< v >$, the correct pronunciation /eI/ is already the most probable, but the word $< have >$ causes the pronunciation still to be uncertain. Trie search proceeds by adding the grapheme immediately to the left of the $< a >$, $< h >$, the grapheme two positions right from $< a >$, $< e >$, and finally the $< e >$ at two positions left from $< a >$. At this bottom level, the pronunciation of $< a >$ in the context of $< ehave >$ is unambiguously /eI/.

The order in which the context graphemes are added to the trie search is not randomly determined, but is computed using the concept of *Information Gain* (IG). This ordering method is used in a similar way in ID3-learning [7]. The main difference with ID3-learning is the fact that our model computes the expansion ordering only once for the complete trie, whereas in ID3-learning the ordering is computed at every node. IG of context graphemes is computed by regarding the training set as an information source capable of generating with a certain probability a number of messages (phonemes) given a string of graphemes. The information entropy of such an information source can be compared in turn for each position in the grapheme string to the average information entropy of the information source when the value of the grapheme at that position is known. The difference is the IG value for that position, and can be interpreted as being the relative *importance* of that position in the context. During trie search, the most important context graphemes are taken first for expansion deeper into the trie, since the context grapheme with the highest IG value is most likely to disambiguate the pronunciation of the grapheme. Figure 2 displays a typical IG value contour, the most important grapheme naturally being the focus grapheme, with decreasing IG values for the graphemes further to the left and right, and right context being slightly more important than its equivalent left context. The data in Figure 2 are the IG values for the complete English CELEX corpus described in the next section.

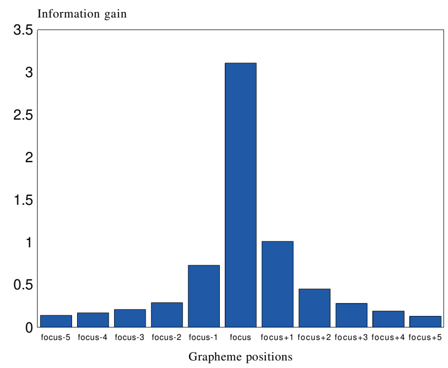

Figure 2. *IG values for ten graphemes surrounding the focus grapheme, computed for the complete English CELEX corpus.*

IBL [1] is a framework and methodology for incremental supervised machine learning. Algorithms developed within this framework are inspired by statistical pattern recognition, especially the rich research tradition on the nearest-neighbour decision rule (see e.g. Devijver & Kittler [5] for an overview) and can be categorised in the class of Similarity-Based Reasoning (SBR) techniques. During training, a memory base is incrementally built consisting of *exemplars*, which in the case of grapheme-to-phoneme mappings consist principally of a string of graphemes (one focus grapheme surrounded by context graphemes), and the associated phonemes and their distribution (as there may be more phonemic mappings to one graphemic string). During testing, a test pattern (a graphemic string) is matched against all exemplars. If the test pattern is in memory, the category with the highest frequency associated with it is used. If it is not in memory, all memory items are sorted according to the similarity of their pattern to the test pattern. The (most frequent) phonemic mapping of the highest ranking exemplar is then predicted as the category of the test pattern. Daelemans & Van den Bosch [4] extended the basic IBL algorithm by introducing Information Gain as a means to assigning different weights to different grapheme positions when computing the similarity between training and test patterns (instead of Euclidean distance).

In the experiments discussed below, we combine the Trie Search algorithm with the IG-aided SBR techniques from IBL. Trie Search succeeds only when a completely matching path can be found up to the node where the phonemic mapping becomes unambiguous. New, unseen test words may very well contain graphemic strings that are not present in the training data. In those cases, Trie Search will fail somewhere halfway. We already suggested that the trie might contain at any node the *most probable* phonemic mapping at that point (or at a fixed contextual width), so that it will still produce a 'best guess' when failing. In an earlier version of our system, we used a separate default table to produce this guess.

The model presented below uses IG-aided SBR on a memory base of exemplars when Trie Search fails. Although this method introduces the relatively costly technique of similarity-based matching, it can be expected to be superior in terms of generalisation performance on new test material to the method of making default 'best guesses'. This hypothesis is tested in the following section.

APPLICATIONS

Experiments were run on two large corpora of word-pronunciation pairs, viz. an English corpus of 56,590 wordforms with their pronunciations, extracted from the CELEX English wordform data base, and a Dutch corpus of 70,000 words with their pronunciations.

In order to automatically construct the trie and the memory base, the phonemic data in the corpus had to be aligned to the graphemic data. The Trie Search algorithm as well as the SBR algorithm presuppose a one-to-one relation between graphemes and phonemes, whereas both in English and in Dutch, there are many cases where a cluster of graphemes maps to one phoneme (especially graphemic vowel combinations). In those cases, the first grapheme of that cluster is mapped to the phoneme, and the other graphemes are mapped to *phonemic nulls*.

In Van den Bosch & Daelemans [2], we showed that both the lookup table (augmented with a default table containing the most frequently occurring phonemic mapping) and the IBL technique (augmented with IG weighing) performed better in terms of generalisation performance than the connectionist NetTalk architecture [8] applied to Dutch data. Testing on data proposed by Nunn & Van Heuven [6], we also investigated how the performance results of the simple Table Lookup model would relate to the results of the knowledge-based MORPA-CUM-MORPHON system reported in [6]. The test data consisted of 1,971 words from newspaper text, compounds, neologisms and low-frequency words. Results show that the Table Lookup model scores significantly higher.

Model	Generalisation Accuracy on Words
Table Lookup	89.5
MORPA-CUM-MORPHON	85.3

We designed the second experiment to determine whether Trie Search (being equivalent to Table Lookup, but with more compression of the stored data) could further be optimized, as sketched in the previous section. To this purpose, we experimented with three versions of Trie Search:

1. Trie Search combined with default tables of fixed contextual width (equivalent to the Table Lookup model described in Experiment 1),
2. Trie Search combined with information on each node on the most probable phonemic mapping at that point in the trie, and
3. Trie Search combined with SBR.

We designed a 10-fold CV experiment series on a subset of the English CELEX-corpus consisting of 10,000 words. In each of the partitions, 10% of the subset was used as training data, and tests were done on the other 90%. The results show a clear advantage of the defaults-on-nodes variant over the fixed default tables variant; optimal results, however, are obtained with the Trie Search + SBR combination. All pairwise comparisons between model scores on both words and phonemes are significant ($p < .001$).

Model	Generalisation Accuracy	
	on Words	on Phonemes
Trie + fixed defaults	20.1	82.0
Trie + defaults on nodes	24.4	83.5
Trie + SBR	28.2	84.4

We designed a second 10-fold CV series of experiments focusing on the Trie Search + SBR model, based on the complete English CELEX-corpus. In each of the partitions, 90% of the corpus was used as training data, and 10% as test data. Averaged over the 10 experiments, the model was able to convert 97.4% of all phonemes (including phonemic nulls) correctly (83.7% on words).

REUSABILITY

We have already demonstrated the application of the Table Lookup / Trie Search technique on two large corpora of Dutch and English, proving language-independency and reusability of the technique. A direct comparison between the two models in terms of generalisation performance is not appropriate given the inherent differences in corpus size, however. When corpus sizes are comparable, application of the technique renders models of which the differences can reveal interesting differences between the languages of the two corpora.

We applied the Table Lookup approach to corpora of equal size (20,000 words) of English (the NetTalk corpus as used in [8]), French (a subset of the Brulex data base [3]) and Dutch (a subset of the corpus used in Experiment 1). After construction, the English table contains 35,000 patterns, the Dutch 27,000 and the French 18,000, reflecting differences in *deepness* of orthography between the three languages. Performance accuracy on a test set (7.5% of the data set), with the inclusion of fixed-length default tables, is 90.1% for the English model, 97.0% for the Dutch model and 98.2% for the French model. In Figure 3, bars indicate the number of patterns that disambiguate between mappings at a certain context width (e.g. 1-1-2: 1 left context grapheme, 2 right context graphemes).

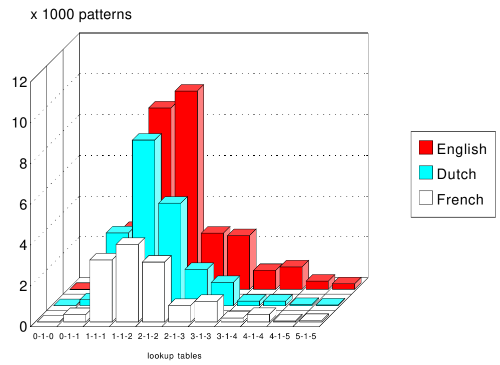

Figure 3. *Table magnitudes of subtables of English, Dutch and French models.*

CONCLUSION

In this paper we have shown that, at least for one linguistic task, there is an alternative to incorporating extensive linguistic knowledge into a linguistic problem solving system. We have shown that for Dutch, TabTalk performs better than both the connectionist backpropagation approach and a state-of-the-art, linguistically sophisticated, knowledge-based system, and that the approach is easily reusable for any language for which a corpus of word-pronunciation pairs exists.

Acknowledgements

We would like to thank Henk Kempff (KUB), Alain Content (ULB), Terrence Sejnowski (UCSD), and CELEX for making available for research purposes the Dutch, French and English data we used. We are grateful to Josée Heemskerk, Anneke Nunn, Gert Durieux, Steven Gillis, Ton Weijters, and Jean Vroomen for comments and ideas.

REFERENCES

[1] Aha, D., D. Kibler, & M. Albert (1991). Instance-Based Learning Algorithms. *Machine Learning* 6, 37-66.

[2] Bosch, A. van den & W. Daelemans (1993). Data-oriented methods for grapheme-to-phoneme conversion. *Proceedings of European Chapter of ACL*, Utrecht, 45-53.

[3] Content, A., P. Mousty, & M. Radeau (1991). Brulex: une base de données lexicales informatisée pour le français écrit et parlé. *L'Année Psychologique*, **90**, 551-566.

[4] Daelemans, W. & A. van den Bosch (1992). Generalization performance of backpropagation learning on a syllabification task. In M. Drossaers & A. Nijholt (Eds.), *Proceedings of the 3rd Twente Workshop on Language Technology.* Enschede: Universiteit Twente, 27-37.

[5] Devijver, P.A. & J. Kittler (1982). *Pattern recognition. A statistical approach.* London: Prentice-Hall.

[6] Nunn, A. & V.J. van Heuven (1993). MORPHON, lexicon-based text-to-phoneme conversion and phonological rules. In V.J. van Heuven & L.C.W. Pols (Eds.), *Analysis and synthesis of speech; strategic research towards high-quality text-to-speech generation.* Berlin: Mouton de Gruyter.

[7] Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning.* **1**, 81-206.

[8] Sejnowski, T.J. & C.R. Rosenberg (1987), Parallel networks that learn to pronounce English text. *Complex Systems*, **1**, 145-168.