Analysing Orthographic Depth of Different Languages Using Data-Oriented Algorithms

Antal van den Bosch*, Alain Content†, Walter Daelemans‡, and Beatrice de Gelder**,†

(Appears in the Proceedings of the 2nd International Conference on Quantitative Linguistics, Moscow, September 1994)

* email: antal@cs.rulimburg.nl
Department of Computer Science,
University of Limburg, Maastricht
PO Box 616
NL-6200 MD Maastricht
The Netherlands
phone +31.43.882018, fax +31.43.252392

† email: acontent@ulb.ac.be
Laboratoire de Psychologie Expérimentale,
Université Libre de Bruxelles
Avenue A. Buyl 117
B-1050 Bruxelles
Belgium
phone +32.2.6504227, fax +32.2.6502209

‡ email: Walter.Daelemans@kub.nl
Institute for Language Technology and AI,
Tilburg University
PO Box 90153
NL-5000 LE Tilburg
The Netherlands
phone +31.13.663070, fax +31.13.662537

** email: B.deGelder@kub.nl
Department of Psychology
Tilburg University
PO Box 90153
NL-5000 LE Tilburg
The Netherlands
phone +31.13.662167, fax +31.13.662370

Summary

We propose a quantitative operationalisation of the concept of orthographic depth, which plays a crucial role in psycholinguistic modelling of reading aloud (and learning to read aloud) in different languages. The orthographic depth of a language is expressed by measuring the complexity of letter-phoneme alignment and the complexity of grapheme-phoneme correspondences within that language. We present the alignment problem and the correspondence problem as tasks to three different data-oriented learning algorithms, and submit them to English, French and Dutch learning and testing material. Generalisation performance metrics are used to propose for each corpus a two-dimensional orthographic depth value.

1. Introduction

There exist substantial differences between alphabetic, syllabic, and logographic writing systems (also referred to as *orthographies*) with respect to their relation between spelling and phonology. Within psycholinguistics, a growing interest is seen in comparing reading processes across orthographies (see Katz and Frost, 1992). Moreover, within the group of alphabetic orthographies, distinct degrees in the complexity of the mapping between orthographic and phonological representations have been suggested. The descriptive notion of *orthographic depth* is coined to characterise the degree of complexity of this mapping (Liberman *et al*, 1980). The orthographic depth of an alphabetic orthography indicates the degree to which it deviates from simple one-to-one letter-phoneme correspondence. Orthographies which have more complex letter-phoneme relations are referred to as deeper orthographies. Examples of deep orthographies are the Hebrew and English writing systems; Serbo-Croatian and Italian are examples of shallow orthographies.

In more detail, orthographic depth can be considered as the composition of at least two separate components. One of these relates to the complexity of the relations between the elements at the graphemic level (graphemes) to those at the phonemic level (phonemes), the issue being how to convert graphemic strings (words) to phonemic strings. Note that our definition of *grapheme* is 'a letter or a cluster of letters that is realised in the phonological transcription as a single phoneme'. The other component relates to the diversity at the graphemic level, and to the complexity of determining the graphemic elements of a word (*graphemic parsing*), or, alternatively formulated, how to align a phonemic transcription to its spelling counterpart. There are differences among languages with respect to the graphemes which are allowed and which are used. These differences are governed by language-specific graphotactic, syllabic and morphological constraints (Klima, 1972; Liberman *et al.*, 1980; Scheerer, 1986).

Based on the (disputed) view that reading aloud involves two independent processes, viz. direct word pronunciation using lexical retrieval, and rule-based grapheme-to-phoneme conversion (i.e., the dual-route model, Coltheart, 1978) cross-linguistic experiments seem to indicate that the balance between these two processes varies as a function of the orthographic depth of the language. More specifically, several authors claim that in shallow orthographies, such as Serbo-Croatian, the analytic rule-based route, operating on grapheme-phoneme correspondences (GPCs), is used more intensively than the lexical retrieval route (see, e.g., Frost *et al.*, 1987). The rationale behind this claim is that using the GPC-based route in a language with a shallow orthography renders more reliable pronunciations than using the rule-based route in the case of a deep orthography, in which the general or default GPCs of the language are often overruled by exceptions. In the latter case, the speaker has to rely to a larger extent on knowledge of whole-word pronunciations.

Thus far, the notion of orthographic depth has only informally been dealt with (e.g., Coltheart, 1978; Katz & Frost, 1992; Carello *et al.*, 1992); clearly, multi-lingual research could benefit from a precise operationalisation. Carello *et al.* (1992) tentatively claim that comparing rule-based GPC systems of two languages may reveal differences in orthographic depth: Serbo-Croatian is likely to have a much smaller GPC set than, for example, English. Coltheart *et al.* (1993) describe a model in which a GPC set for English is learned from examples by a learning algorithm. Automatic, data-oriented learning algorithms seem to provide an appropriate means for extracting statistical facts from language data related to orthographic depth, without incorporating any linguistic bias in the form of language-specific constraints or heuristics. Daelemans and van den Bosch (1993) demonstrate that the application of data-oriented techniques to morpho-phonological domains, such as grapheme-to-phoneme conversion, is language-independent, and can be done for any language for which a computer-readable corpus exists.

In this paper, we investigate whether the application of three different data-oriented learning algorithms to three alphabetic orthographies, viz. English, French and Dutch, reveals any differences in orthographic depth among these three languages. To this purpose, one algorithm is trained on the domain of graphemic parsing (Section 3.1), and the two remaining algorithms are trained on grapheme-to-phoneme conversion (Sections 3.2 and 3.3).

2. Corpus Selection

We have extracted our training and testing material from three computer-readable corpora of English, French and Dutch, all consisting of large lists of word spelling-pronunciation pairs. In the case of English, we used the *NETtalk* corpus of American English, first used by Sejnowski and Rosenberg (1987); the French material was extracted from the *Brulex* corpus (Content *et al.*, 1990); the Dutch material was extracted from a large Dutch lexical data base. To ensure experimental validity, we obtained a close similarity between these corpora by restricting their size to about 20,000 words for each corpus.

Our general experimental method involves the application of an automatic data-oriented learning algorithm to a fixed amount of learning (training) material, and the testing of the generalisation ability of the learned model using a fixed amount of new testing material. To this purpose, the three language corpora were split into training and test sets which remained fixed throughout all experiments. Each corpus was partitioned into a 1/13 test set (7.7% of the data set) and a 12/13 training set. This is an arbitrary partition. However, it should be noted that the focus of our experiments is on comparing performance results rather than on optimizing performance.

The training sets thus obtained consist of a large number of word-pronunciation pairs, for example, the English pair <shoe> - /ʃu/. To be able to convert the 4-letter string <shoe> to the two-phoneme pronunciation /ʃu/, a system has to solve two subproblems: (i) that the string <shoe> contains two graphemes, <sh> and <oe>, and (ii) that <sh> maps to /ʃ/, and <oe> maps to /u/ in this particular context. The knowledge needed for (i) is part of knowing which letter clusters can occur in a language; for (ii), it is needed to know what the possible correspondences between graphemes and phonemes within a language are. These two subproblems of converting spelling to pronunciation correspond to what was referred in the first section as the two most important components of orthographic depth, i.e., subproblem (i) relates to complexity at the graphemic level, and subproblem (ii) relates to the complexity of the relation between the graphemic and the phonemic level. Furthermore, (ii) subsumes having solved (i).

Our experiments focus on analysing the complexity of (i) and (ii) separately. We present the two subproblems as tasks to our learning algorithms. For task (i), we train a learning algorithm on the spelling-pronunciation pairs of the three training corpora. For task (ii), we simulate the situation where (i) has already successfully been solved, and train two different learning algorithms on converting graphemic words to their phonemic transcription. In the case of English, these graphemic parsings were available: in the NETtalk corpus, the phonemic strings are supplied with *phonemic nulls*, which are inserted at points where in the spelling string a graphemic letter cluster occurs. For example, the phonemic transcription of <shoe>, /ʃu/, is aligned to fit the four-letter spelling word as /ʃ-u-/. Although indirect, this alignment, containing two phonemic nulls, indicates that the word <shoe> contains two graphemes. The same kind of alignment was performed for the Dutch and French corpora using pattern-matching algorithms and hand-correction. Clearly, these algorithms and corrections introduce linguistic knowledge in a supposedly language-independent framework. A fully language-independent and linguistic knowledge-independent system would perform both (i) and (ii), using the graphemic analysis in (i) as input to subsystem (ii). In fact, Daelemans and van den Bosch (1994) demonstrate a data-oriented, language-independent system which successfully integrates two high-performance data-oriented learning algorithms performing (i) and (ii) in sequence. In this paper, we focus on a separate analysis of the two subproblems.

3. Three Learning Algorithms

3.1. Grapheme-Phoneme Correspondences Extraction

Graphemic parsing of a spelling word primarily implies knowing which are the possible and typical graphemes in a language. The Grapheme-Phoneme Correspondences Extraction (henceforth GPCE) model described here is trained to capture this knowledge by an automatic, data-oriented learning algorithm. Rather than being trained explicitly on parsing a spelling string into graphemes, the GPCE model is aimed at constructing a memory base of hypothesised grapheme-phoneme mappings, so that, after training, the GPCE model is able to express probabilistic scores of a given graphemic analysis of a spelling word. The GPCE algorithm, in its most basic form, takes a training corpus of word-pronunciation pairs, and constructs on the basis of that corpus a memory base, containing all occurring grapheme-phoneme correspondences within that corpus. The construction algorithm of the GPC base has no knowledge of typical or regular grapheme-phoneme mappings: therefore some of the hypothesised correspondences will be linguistically inappropriate. To obtain this memory base of mappings, or rather *Grapheme-Phoneme Correspondence exemplars*, the following steps are taken for all word-pronunciation pairs in the training corpus:

(a) For each word-pronunciation pair, generate all possible parsings of the word in as much segments as there are phonemes (i.e., generate all possible letter clusters that can map onto one phoneme). For example, the French word <chat> (cat), with pronunciation /ʃɑ/, results in three parsings: <cha|t>, <ch|at>, and <c|hat> (the '|' indicates the inserted parsing boundary).

(b) For each of the generated parsings, map each segment in that parsing to the corresponding phoneme. In the example of <chat>, this results in 6 GPC exemplars, two of which are correct (marked *): <cha>-/ʃ/, <ch>-/ʃ/ (*), <c>-/ʃ/, <t>-/ɑ/, <at>-/ɑ/ (*), and <hat>-/ɑ/.

(c) Store each derived GPC exemplar in the GPC base. If it is already stored, increase the occurrence field of the GPC exemplar, and update the occurrence of the phonemic mapping (or create a new phonemic mapping field if the phonemic mapping was not encountered earlier). If it is not present in the GPC base, create a new exemplar, and initialise its occurrence field.

After training, a memory base is available which consists of a large number of hypothesised GPC exemplars. The occurrence field of each of these GPC exemplars simply expresses the absolute number of occurrences of the GPC exemplar in the training corpus. The magnitude of this number is relative to two factors: (i) the size of the grapheme (single letter graphemes are encountered more frequently than multi-letter graphemes), and (ii) the 'typicality' of the grapheme, which is a vague notion but which generally captures the graphotactics (i.e., which letter combinations are permitted, and which are impossible or odd) of the orthography. The algorithm described thus far has no direct relation with the problem of graphemic parsing of which we want to investigate the complexity for the English, French and Dutch corpora. However, the fuzzy knowledge about the typicality of graphemes can be used to estimate the most probable graphemic parsings for new test words. To obtain these estimates, the following algorithm is used: for each new test word,

(a) generate all possible graphemic parsings. At one extreme, a parse is generated which takes each letter as a separate grapheme; at the other extreme a parse contains only graphemes of maximal length (e.g., 4 letters in English, as in <ough>);
(b) for each graphemic parsing, search the GPC exemplar base for all matching GPC exemplars. Each parsing is given a score which is the sum of the occurrences of all matching GPC exemplars;
(c) the parsing with the highest score is taken as output.

Given the analysis already present in the prepared corpus, it can be determined for each test word if the graphemic parsing is correct. This model feature is examined in Section 4.1.

3.2. Decision Tree Learning

A detailed description of the Decision Tree Learning (also referred to as Trie Compression) and Decision Tree Search algorithms can be found in Van den Bosch and Daelemans (1994, to appear). The Decision Tree model converts spelling to phonology using letter-phoneme correspondence chunks, which are stored as paths in a Decision Tree structure, and which are extracted from a training corpus of (aligned) word-pronunciation pairs. Each letter-phoneme correspondence chunk that is stored consists of a focus letter, a number of left and right context letters and an associated phonemic mapping (i.e., the phoneme *or phonemic null* to which the focus letter maps). The stored context may vary from being empty to containing whole words: it is exactly the minimal context in which the letter-phoneme mapping is unambiguous. An empty context occurs when dealing with, for example, the French letter <ç>, which unambiguously maps to /s/, regardless of the context. In the Decision Tree, this knowledge is stored as a single-node path, with an end node high up in the tree. When a large context is needed, it is stored as a longer path down the Decision Tree. For example, the phonemic mapping /əʊ/ to the first <o> in <photograph> involves a right context of 8 characters (i.e., practically the whole word) to disambiguate it from the phonemic mapping /ə/ to the first <o> of <photography>. The more irregular a letter-phoneme correspondence is, the deeper the mapping is stored in the Decision Tree.

The knowledge present in a training corpus is stored in the tree in such a way that no grapheme-phoneme correspondence information is lost. The information inherent in the training corpus is simply compressed; the amount of memory needed to store this information is minimised. Effectively, the Decision Tree is a compressed word-pronunciation corpus from which the correct pronunciation for any word in the training set can be retrieved. This is done by finding for each letter in its specific context a matching path in the tree leading to its phonemic mapping. However, this may not be the case for a test word which might contain substrings not encountered in the training corpus. When Decision Tree Search is performed with such a string, the retrieval algorithm will not be able to find an exactly matching path, and consequently will not retrieve unambiguous phonemic information. The model attempts to solve this problem by storing at every tree node information about the *most probable* phonemic mapping at that node. When Tree Search fails, this extra information enables the model always to suggest a 'best guess', a property of the model essential for optimal generalisation performance.

For each of the three language corpora, the amount of compression compared to the original training material as well as the performance-accuracy scores on test material will be examined more closely in Section 4.2.

3.3. Similarity-Based Reasoning

The Similarity-Based Reasoning (SBR) model attempts, just as the Decision Tree Learning model, to store letter-to-phoneme correspondence knowledge in such a way that it can be successfully used to retrieve the phonemic transcription of new, previously unencountered test words.

During training of the SBR model, a *memory base* is constructed consisting of letter-to-phoneme instances, called *exemplars.* To do this, each word in the training corpus is converted into a number of letter string patterns. Each pattern consists of a focus letter surrounded by a fixed number of left and right context letters, together with the corresponding phoneme of the (aligned) phonemic transcription. For our models, we kept the number of left and right context letters at 5. Patterns are stored as exemplars in the memory base; whenever a duplicate letter-string pattern is found, the occurrence count of the phonemic mapping of the stored exemplar is increased, or a new phonemic mapping field is added to the exemplar if that phonemic mapping was not encountered earlier. To retrieve the phonemic transcription of a test word, it is converted into the same fixed-length letter-string patterns. Each of these patterns is matched against all memory exemplars. If the test pattern matches an exemplar, the phonemic category with the highest frequency associated with the exemplar is retrieved. 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.

A more detailed description of the SBR algorithm can be found in Daelemans and van den Bosch (1992).

4. Results

4.1. Grapheme-Phoneme Correspondences Extraction

The GPC memory base construction algorithm has been applied to training sets of French, English and Dutch which are a subset of the original training set. Each training set contained 5,000 words. These smaller sets were chosen, because pilot experiments showed a performance convergence at data set sizes above approximately 1,000 words. After construction, the full test corpus was processed through the GPC test algorithm. From the resulting best guessed graphemic analyses and phonemic mappings, performance scores were computed expressing the percentage of incorrect graphemic analyses of words. Table 1 lists these figures for the three languages.

corpus	% incorrectly aligned words
English	75.5
French	87.1
Dutch	78.7

Table 1. *Percentage of incorrectly aligned test words obtained with the three GPCE models after memory base construction, trained on 5000-word partitions of the original training sets and tested on the full test sets.*

Obviously, the performance scores listed in Table 1 are not high. This is due to the overall fuzziness of the GPCE model, as it is mainly concerned with finding *regular* graphemes rather than exceptions. Nevertheless, the differences between the three corpora are apparent. In the case of the English corpus, alignment is relatively less complex than in the cases of the Dutch and French corpus. In terms of correctly aligned test words, the French model clearly renders the least accurate results. In other words, the GPCE model trained on the French material shows worse generalisation capabilities than the models trained on Dutch and English, while being trained on an identical amount of training material. From these results, it can be concluded that graphemic parsing is more complex in French than in Dutch or English.

4.2. Decision Tree Search

The application of Decision Tree Learning to the three training corpora resulted in three models of very different size. Since Decision Tree Learning is based on removing redundancy from a corpus by compressing the knowledge without losing any (i.e., *lossless compression*), higher compression indicates that the corpus contains more regularity. In terms of compression of memory usage, the French model was compressed by a factor of 90.8%, the Dutch model by 87.4%, and the English model by 70.9%. The English material appears to contain less redundancy, and can be regarded as more irregular than the French and Dutch data. The performance on the test words provides more clues concerning differences between English on the one hand and French and Dutch on the other. Table 2 lists the generalisation performance of the three models on the test material.

language	% incorrect words	% incorrect mappings
English	45.7	9.0
French	10.9	1.7
Dutch	18.6	2.4

Table 2. *Generalisation performance (on test data) of the three models. Scores listed on incorrectly produced words and incorrectly transliterated letters.*

Best performance scores are obtained with the French model. In terms of correctly transliterated words, the Dutch model scores significantly lower, but in terms of correctly converted phonemes (the most unbiased measure), the scores are roughly similar. The English model scores notably worse than the French and Dutch model on both words and phonemes.

Figure 1 presents another view on the differences between the three automatically constructed Decision Trees. In this Figure, bars indicate the number of stored paths that end at a certain context width. The labels on the x-axis indicate this context. For example, the largest white bar in the front row, labelled '1-1-2', indicates that, in the French model, most letter-phoneme correspondence chunks use a context of one left context letter, and two right context letters.

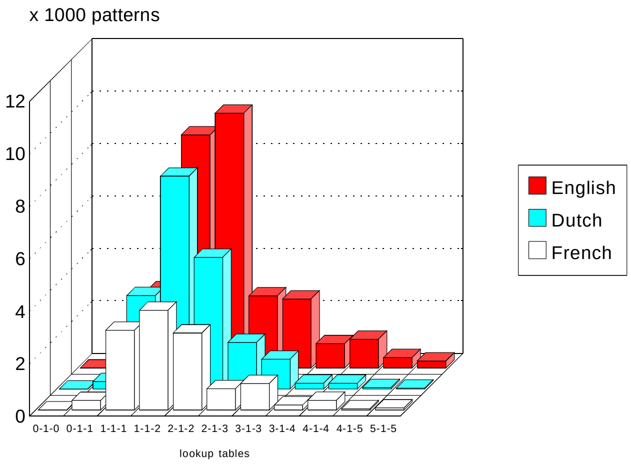

Figure 1. *Numbers of end nodes, represented by bars, per tree depth (path length), for three SPC models trained on comparable corpora of English, Dutch and French.*

4.3. Similarity-Based Reasoning

As described earlier, the SBR memory base is constructed for each corpus by converting all word-pronunciation pairs into fixed-length letter-string patterns, which were then stored as exemplars in the memory base. Since there were not many duplicate 5-1-5 patterns in any of the three corpora, large memory bases resulted. For example, in the case of English, out of the 135,406 5-1-5 patterns, 120,062 exemplars were stored (11.3% compression). For Dutch, compression was 12.0% (156,449 exemplars stored), and for French 17.8% (129,054 exemplars stored), indicating that the French corpus contains more partly similar words than the other two corpora.

Table 3 displays the generalisation accuracy on test words and phonemes for the three models. The results show high scores for Dutch and French, and a significantly lower score for English, especially when expressed in the percentage of correctly transliterated words. The performance results are highly similar to those obtained with the Decision Tree Search models.

language	% incorrect words	% incorrect mappings
English	45.9	9.0
French	11.0	1.7
Dutch	17.5	2.2

Table 3. *Generalisation accuracy on test words and mappings by the three Similarity-Based Reasoning models.*

5. Conclusions

The application of three data-oriented machine learning techniques on three grapheme-to-phoneme corpora has revealed differences between the orthographic complexity within these corpora. In line with the propositions of Klima (1972) and Liberman *et al.* (1980), we propose that the problems of graphemic alignment and grapheme-to-phoneme conversion are the basic components of converting spelling words to their phonemic transcription. Although they are not totally independent problems, they can be regarded as the two most distinct components, or, geometrically speaking, *dimensions* in the space describing the complexity of an orthography.

We argued earlier that the first dimension of orthographic depth, the complexity of graphemic analysis (i.e., the problem of aligning phonemic strings to letter strings), is embedded in the memory base of the GPCE model, and is expressed in the error output of the model when applied to unseen test words. Table 1 displays the difference obtained between the GPCE models of the three language corpora. We propose to take this measure indicating the number of incorrectly aligned words to express the complexity of dimension (i) of orthographic depth. It should be stressed again that the absolute magnitude of the measures is not important here: the key importance lies in the *relative differences* between the three languages.

The complexity of converting strings of graphemes to strings of phonemes is, amongst other metrics such as Decision Tree Learning compression factors, Decision Tree sizes and SBR memory base compression factors, most prominently expressed in the generalisation accuracy on the transliteration of letters to phonemes in test words. Furthermore, Decision Tree Search and SBR performances are highly similar (see Tables 2 and 3). We propose to take this performance measure as a measure of complexity of going from the level of graphemes to the level of phonemes, i.e., dimension (ii) of orthographic depth: the higher this number is, the more complex the problem is within a certain corpus. Again, only the relative differences between the three languages matter here.

The two dimensions and the three points marking the three corpora are displayed graphically in Figure 2, constituting a 'map' in which the relative distance of the three corpora within the two-dimensional orthographic depth space is clearly expressed.

Our data-oriented, generic, two-dimensional classification of the complexity of grapheme-to-phoneme conversion can be used as a platform for determining an unbiased grounding of orthographic depth for any corpus in any language. The only restriction the corpus must adhere to at this point is the approximate number of words. A number of approximately 20,000 words, we would like to argue, is sufficiently large to capture practically all occurring graphemes and letter-phoneme mappings of a language, i.e., enough to ensure that the learning algorithms are confronted with every regularity and irregularity of the orthography that they are confronted with.

Acknowledgements

This research was partly supported by a grant from the Human Frontier of Science Programme *Processing consequences of contrasting language phonologies.* During this research, the first author was affiliated with the Institute for Language Technology and AI (ITK) at Tilburg University, the Department of Psychology at Tilburg University, and the Experimental Psychology Lab at the Université Libre de Bruxelles. We would like to thank Terrence Sejnowski and Henk Kempff for making available the NETtalk corpus and the Dutch corpus, respectively. Thanks are also due to Jaap van den Herik for his valuable comments on the text.

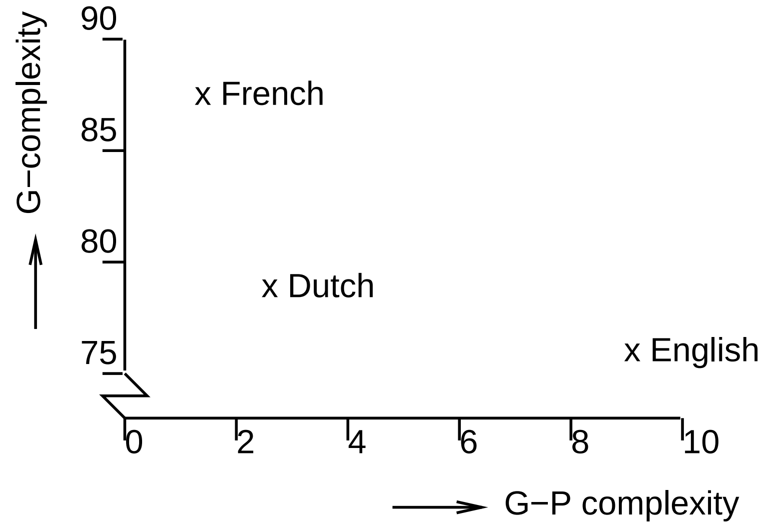

Figure 2. *Graphical display of the two-dimensional orthographic depth space, with 'x's marking the three corpora.*

References

Carello, C., Turvey, M., Lukatela, G. (1992). "Can theories of word recognition remain stubbornly non-phonological?" In *Haskins Laboratories Status Report on Speech Research* (pp. 193–204). Haskins Laboratories.

Coltheart, M. (1978). "Lexical access in simple reading tasks." In G. Underwood (Ed.), *Strategies of Information Processing* (pp. 151–216). London: Academic Press.

Content, A., Mousty, P., Radeau, M. (1990). "Brulex: Une base de données lexicales informatisée pour le français ecrit et parlé." *L'Annee Psychologique*, 90, 551–566.

Daelemans, W., van den Bosch, A. (1992). "Generalisation performance of backpropagation learning on a syllabification task." In M. Drossaers & A. Nijholt (Eds.), *TWLT3: Connectionism and Natural Language Processing*: Enschede: Twente University.

Daelemans, W., van den Bosch, A. (1993). "Tabtalk: Reusability in data-oriented grapheme-to-phoneme conversion." In *Proceedings of Eurospeech '93* (pp. 1459–1466).: Berlin: T.U. Berlin.

Daelemans, W., van den Bosch, A. (1994). "A language-independent, data-oriented architecture for grapheme-to-phoneme conversion." In *Proceedings of ESCA-IEEE'94*.

de Gelder, B. (1993). "Reading acquisition: The rough road and the silken route." Paper to appear in the Journal of Chinese Linguistics.

Frost, R., Katz, L., Bentin, S. (1987). "Strategies for visual word recognition and orthographical depth: a multilingual comparison." *Journal of Experimental Psychology: Human Perception and Performance*, 13, 104–115.

Katz, L., Frost, R. (1992). "The reading process is different for different orthographies: the orthographic depth hypothesis." In *Haskins Laboratories Status Report on Speech Research 1992* (pp. 147–160). Haskins Laboratories.

Klima, E. (1972). "How alphabets might reflect language." In J. Kavanagh & I. Mattingly (Eds.), *Language by Ear and by Eye: The Relationship Between Speech and Reading* (pp. 57–80). Cambridge, MA: The MIT Press.

Liberman, I., Liberman, A., Mattingly, I., Shankweiler, D. (1980). "Orthography and the beginning reader." In J. Kavanagh & R. Venezky (Eds.), *Orthography, Reading and Dyslexia* (pp. 137–153). Baltimore: University Park Press.

Scheerer, E. (1986). "Orthography and lexical access." In G. Augst (Ed.), *New trends in graphemics and orthography* (pp. 262–286). Berlin: De Gruyter.

Sejnowski, T. J., Rosenberg, C. S. (1987). "Parallel networks that learn to pronounce English text." *Complex Systems*, 1, 145–168.

van den Bosch, A., Daelemans, W. (1993). "Data-oriented methods for grapheme-to-phoneme conversion." In *Proceedings of the 6th Conference of the EACL* (pp. 45–53).