Journal of Quantitative Linguistics 1, 3, 178-188, 1994

# Measuring the complexity of writing systems

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

* Department of Computer Science, University of Limburg, e-mail: antal@cs.rulimburg.nl; † Laboratoire de Psychologie Expérimentale, Université Libre de Bruxelles, e-mail: acontent@ulb.ac.be; ‡ Institute for Language Technology and AI, Tilburg University, e-mail: Walter.Daelemans@kub.nl; ** Department of Psychology, Tilburg University, e-mail: B.deGelder@kub.nl.

To appear in the Journal of Quantitative Linguistics.

## Summary

We propose a quantitative operationalisation of the complexity of a writing system. This complexity, also referred to as orthographic depth, plays a crucial role in psycholinguistic modelling of reading aloud (and learning to read aloud) in several languages. The complexity of a writing system is expressed by two measures, viz. that of the complexity of letter–phoneme alignment and that of the complexity of grapheme–phoneme correspondences. 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 writing system complexity value.

## 1. Introduction

Substantial differences exist 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 writing systems (cf. Katz and Frost, 1992). Moreover, within the group of alphabetic writing systems, 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 writing system indicates the degree to which it deviates from simple one-to-one letter–phoneme correspondence. Writing systems with 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).

A potential third component of orthographic depth which we do not take into consideration in the present work, is the assessment of the extent to which the spelling of a word provides information not included in the phonemic representation. French has often been quoted as a prototypical example of a language with a arbitrary and inconsistent *spelling.* This is due to the fact that in many cases in French, a phoneme admits several graphemic representations (for example, /ãː/ is the phonemic mapping of the graphemes <in>, <ein>, <ain>, <aint>, <ym>, <en>, and <ent>), making it hard for the learner to derive the spelling from the phonemic string by applying simple rules. Note that this potential third component of orthographic depth is irrelevant in a framework in which one is investigating the complexity of performing the conversion from the spelling level to the phonemic level. In this paper, we take the latter approach and disregard the complexity of converting phonemic representations to spelling.

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 (cf. 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 in many cases 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 and 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.

The data-oriented approach furthermore presents an interesting alternative to the traditional linguistic approach to describing the correspondences between spelling and phonology. This traditional approach is based on detailed linguistic expert analyses of writing systems (e.g., Venezky, 1970, for English; Gak, 1976, for French). Apart from the fact that an expert analysis is expensive, and that expert knowledge of a writing system is often hardly reusable for other writing systems, traditional approaches take into account a number of very diverse sources of information. Traditional grapheme-to-phoneme conversion rules may refer to, for example, morphological or etymological knowledge. Our data-oriented approach presents an alternative to this tradition by limiting the levels at which it operates, to strictly the graphemic and the phonemic level, and attempting to extract as much knowledge from these two levels as possible. This approach offers a possible solution to the 'knowledge acquisition bottleneck' that any traditional linguistic expert analysis is confronted with.

In this paper, we investigate whether the application of three different data-oriented learning algorithms to three alphabetic writing systems, 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–transcription pairs (i.e., pairs of spelling words and their phonemic transcriptions). 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 word–transcription pairs for each corpus. No information other than spelling and phonemic transcription was used in the experiments; other word-specific features, such as word frequency, place of word stress, and syllable boundaries, were not included in the corpora.

Our general experimental method involved 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 large numbers of word–transcription pairs, such as in the case of the English corpus, the pair <shoe> - /ʃu/. To be able to convert the four-letter string <shoe> to the two-phoneme transcription /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 three learning algorithms. For task (i), we train a learning algorithm on the spelling–transcription 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 are 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-/. This null insertion serves to indicate that the grapheme <sh> maps to the phoneme /ʃ/, and that the grapheme <oe> maps to the phoneme /u/. The leftmost letter of the grapheme is consistently aligned with the phoneme; other letters to the right are aligned with phonemic nulls. 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 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. The GPCE model is not explicitly trained to parse strings of spelling letters into graphemes. The model assembles in its training phase a large data base of hypothesised grapheme–phoneme correspondences, by extracting them in an unintelligent way from a training corpus of spelling word–transcription pairs. In the testing phase, it uses this data base to compute in an automatic, unbiased way the probability of a proposed graphemic parsing of a word.

The data-base-construction algorithm has no knowledge of typical or linguistically appropriate grapheme–phoneme mappings. Therefore some of the hypothesised correspondences will be linguistically inappropriate. To obtain the data base of mappings, or rather *Grapheme–Phoneme Correspondence exemplars*, the following three algorithmic steps are taken for all word–transcription pairs in the training corpus.

(a) For each word–transcription 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/, is parsed in three ways: <cha|t>, <ch|at>, and <c|hat> (the '|' indicates the inserted parsing boundary between the hypothesised graphemes). Note that the second parsing, <ch|at>, is the linguistically appropriate parsing, and that the other parsings involve hypothesised graphemes that are linguistically inappropriate (i.e., <cha> and <hat>).

(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 appropriate (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. The first factor is the size of the grapheme: the major part of the graphemes hypothesised during the generation of graphemic parsings contains only one letter. In the three corpora under consideration, the majority of the graphemes is in fact formed by single-letter graphemes (many words do not even contain any multiple-letter graphemes). Consequently, multiple-letter graphemes are generated less often. The second factor having influence on the number of occurrences of a certain GPC exemplar is the somewhat vague notion of 'typicality' or linguistic appropriateness of the grapheme. This can be illustrated by looking at the occurrence counts of some hypothesised three-letter graphemes in the French GPCE model. The three-letter grapheme with the highest occurrence count is <ent> (viz. 31,682), which is indeed very often realised in phonemic transcriptions as the single phoneme /ɑ̃ː/. Other three-letter graphemes with relatively high occurrence scores include <sse> (1,985), <nne> (1,686), and <que> (1,377). On global inspection, it appears that hypothesised graphemes with low occurrence scores are often linguistically inappropriate. For example, <int> is hypothesised 798 times, <thé> is hypothesised 152 times, <apl> 3 times. The graphemes <urv> and <lvé> are examples of highly inappropriate graphemes which are hypothesised only once on the basis of the French corpus.

The GPCE 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 noisy knowledge about the typicality of graphemes present in the GPC base can be used to estimate the most probable graphemic parsings for new test words. To obtain these estimates, the following three-step 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 (e.g., in the case of the English word <book>, <b|o|o|k>); at the other extreme a, parse is generated which contains only graphemes of maximal length (e.g., the complete word <book> as a single grapheme, since English graphemes can contain up to 4 letters, as in the grapheme <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 its individual 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 proposed by the model is correct. This model feature is examined in Section 4.1.

### *3.2. Decision Tree Learning and Search*

Our first model which we trained on the task of converting graphemes to phonemes is the Decision Tree model. In this model two algorithms are combined. The first, Decision Tree Learning (also referred to as Trie Compression) is used to construct a decision tree on the basis of a training corpus of grapheme-phoneme correspondence examples (the training material). The second algorithm, Decision Tree Search, is used to retrieve information from the decision tree in order to find the appropriate phonemic mapping to (possibly unseen) graphemic input strings. Detailed descriptions of both algorithms can be found in Daelemans and Van den Bosch (1993) and Van den Bosch and Daelemans (to appear).

The Decision Tree model converts words to their phonemic transcription in a letter-oriented way. For each letter in a spelling word, the model attempts to find the most appropriate phonemic mapping, given the current letter context. To this purpose, the Decision Tree Learning algorithm automatically constructs a decision tree containing letter–phoneme correspondence chunks, which are in fact context-sensitive rewrite rules with no limit on the size of the context. These letter–phoneme correspondences are automatically extracted from the training corpus (consisting of aligned word–transcription pairs), and are stored as paths in the decision tree. 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: the criterion for storing a certain context is that 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.

When a letter–phoneme correspondence chunk is extracted from the training corpus to be inserted in the decision tree, it is converted into the format of a decision tree path by placing the context letters in a fixed order, which reflects their relative importance. This ordering follows from computing the average *Information Gain* for each of the context positions (for a more detailed description of the computation and application of Information Gain, a concept from Information Theory, see Quinlan, 1986; Daelemans and Van den Bosch, 1992; and Van den Bosch and Daelemans, to appear). The Information Gain of a context position can be seen as its average relative importance in disambiguating between the different phonemic mappings of the focus letter, and can be computed automatically. When the Information Gain of different context positions is computed for the three languages, it appears that the relative importance of different context positions is generally the same for the three languages. The results indicate that, on the average, the closer a context letter is to the focus letter, the more important it is in the disambiguation between different phonemic mappings of the focus letter. Furthermore, right context appears to be slightly more important than left context. In Figure 1, the Information Gain values of context letters (up to a context width of 5 left characters and 5 right characters) are shown graphically, combining the results computed for the three languages. From Figure 1, it can be seen that the same fixed importance ordering of context letters can be used for the three languages.

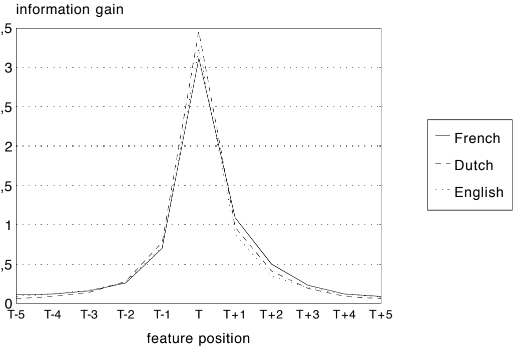

Figure 1. *Information Gain values of different context letter positions, ranging from 5 positions to the left of the focus letter (T−5), to 5 positions to the right of the focus letter (T+5), computed for the three corpora.*

The purpose of using the ordering derived from computing the Information Gain values of context features, is that it minimises the average depth in the decision tree at which an unambiguous mapping is stored. The first context positions investigated by the retrieval algorithm for finding a matching path are the context letters that are known to be the most important ones (on the average). Thus, the size of the decision tree as well as the effort needed to retrieve information in the tree is minimised.

Effectively, the decision tree is a compressed word–transcription corpus from which the correct pronunciation for any word in the training set can be retrieved. Retrieval takes place by finding for each letter in its specific context a matching path in the tree leading to its phonemic mapping. However, this does not succeed for any test word which contains substrings not encountered in the training corpus. When a Decision Tree Search is applied with such a letter string as input, 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 ambiguity by storing at every tree node information about the *most probable* phonemic mapping at that node. When Decision Tree Search fails at some node at a certain depth in the tree, the extra probabilistic information at that node enables the Decision Tree Search algorithm to suggest a 'best guess', a property of the model essential for optimal generalisation performance (see, e.g., Van den Bosch and Daelemans, 1993, for an evaluation of this aspect of the model).

For each of the three language corpora, the amount of compression compared to the original training material as well as the generalisation performance 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 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.* For this construction, 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 three SBR models, we set the number of left and right context letters at 5. As an example, Table 1 lists the 5 patterns that are constructed when processing the word <shoes> (aligned transcription /ʃ-u-z/).

| left context | | | | | focus | right context | | | | | pho- |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 | T | 1 | 2 | 3 | 4 | 5 | neme |
| _ | _ | _ | _ | _ | s | h | o | e | s | _ | ʃ |
| _ | _ | _ | _ | s | h | o | e | s | _ | _ | - |
| _ | _ | _ | s | h | o | e | s | _ | _ | _ | u |
| _ | _ | s | h | o | e | s | _ | _ | _ | _ | - |
| _ | s | h | o | e | s | _ | _ | _ | _ | _ | z |

Table 1. *Example conversion of the word <shoes> - /ʃuz/ into 5 fixed-width patterns as used by the SBR model. The focus letter is in position T.*

Patterns are stored as exemplars in the memory base. Whenever a letter-string pattern has already been stored in the memory base, the occurrence count of the phonemic mapping of the stored exemplar is increased; a new phonemic mapping field is added to the exemplar if the phonemic mapping of the new letter-string pattern 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 similarity metric counts the number of identical letters in identical positions in the test pattern and all exemplars; a co-occurring letter is counted by the value of the Information Gain of the context position it is in. Thus, the similarity matcher of the SBR model prefers exemplars that match in the middle, i.e., around the focus letter, over exemplars that match on the left-hand or right-hand side of the pattern. The (most frequent) phonemic mapping of the highest ranking exemplar is then predicted as the category of the test pattern. The performance results of the SBR model on test material are investigated in Section 4.3.

## 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 correct graphemic analyses of words. Table 2 lists these figures for the three languages.

| | % correctly |
|---|---|
| corpus | aligned words |
| English | 24.5 |
| French | 12.9 |
| Dutch | 21.3 |

Table 2. *Percentage of correctly 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 2 are not high. This is due to the fact that the GPCE model is mainly concerned with finding *regular* graphemes rather than exceptions. The model generally favours graphemic alignments consisting only of single letter-to-phoneme mappings, since one-letter graphemes are hypothesised far more often that multi-letter graphemes. Apart from this observation, it can be seen that there are apparent differences between the three corpora. In the case of the English corpus, alignment is relatively less complex than in the cases of the Dutch and French corpora. 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 Learning and Search*

The application of Decision Tree Learning to the three training corpora resulted in three decision trees of very different size. Since Decision Tree Learning is based on removing redundancy from a corpus by compressing the information on grapheme-phoneme correspondences in the form of paths in a decision tree, 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 thus 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 3 lists the generalisation performance of the three models on the test material.

| language | % correct words | % correct mappings |
|---|---|---|
| English | 54.3 | 91.0 |
| French | 89.1 | 98.3 |
| Dutch | 81.4 | 97.6 |

Table 3. *Generalisation performance on test material of the three Decision Tree models. Scores are listed on correctly transcribed words and correctly transcribed letter–phoneme mappings.*

The best performance scores, in terms of correctly transcribed letters and whole words, are obtained with the French Decision Tree model. In terms of correctly transliterated words, the Dutch Decision Tree model scores somewhat 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 2 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.

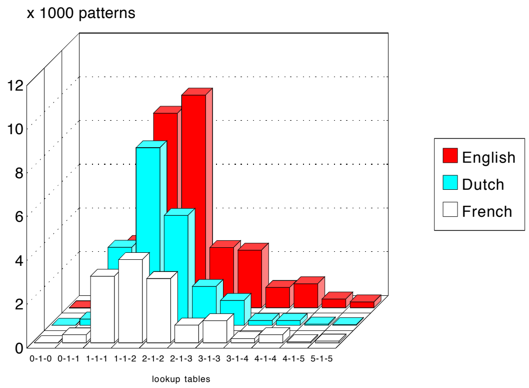

Figure 2. *Numbers of end nodes, represented by bars, ordered by their positions in the tree (i.e., path lengths, indicated by a context width indicators denoting <number of left context characters> –1– <number of right context characters>), for the Decision Tree models trained on the English, Dutch and French corpora.*

### *4.3. Similarity-Based Reasoning*

As described earlier, the SBR memory base is constructed for each corpus by converting all word–transcription 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 4 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 | % correct words | % correct mappings |
|---|---|---|
| English | 54.1 | 91.0 |
| French | 89.0 | 98.3 |
| Dutch | 82.5 | 97.8 |

Table 4. *Generalisation performance on test material of the three Similarity-Based Reasoning models. Scores are listed on correctly transcribed words and correctly transcribed letter–phoneme mappings.*

## 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 complexity of a writing system (its orthographic depth) can be seen as the composition of two components, viz. the complexity of graphemic parsing and the complexity of grapheme-to-phoneme conversion. Although they are not totally independent, they can be regarded as the two most distinct components, or, geometrically speaking, *dimensions* in the space describing the complexity of a writing system.

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 GPC base of the GPCE model, and is reversely expressed in the generalisation performance score of the model when applied to unseen test words. Table 2 displays the difference obtained between the GPCE models of the three language corpora. We propose to take the reverse of these performance scores, i.e. the percentage of *incorrectly* aligned words, as the measure of the complexity of dimension (i) of orthographic depth, under the assumption that the lower the performance, the higher the complexity. 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 language corpora.

The complexity of converting strings of graphemes to strings of phonemes is, amongst other measurable model features 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, the generalisation performance scores of the Decision Tree models and the SBR models are highly similar (see Tables 3 and 4). We propose to take the reverse of these generalisation performance scores as the measure of the complexity of going from the level of graphemes to the level of phonemes, i.e., dimension (ii) of orthographic depth: the lower the generalisation performance score, the more complex the problem. Again, here only the relative differences between the three languages matter. The two dimensions and the three points marking the three corpora are displayed graphically in Figure 3, constituting a 'map' in which the relative distance of the three corpora within the two-dimensional orthographic depth space is clearly expressed.

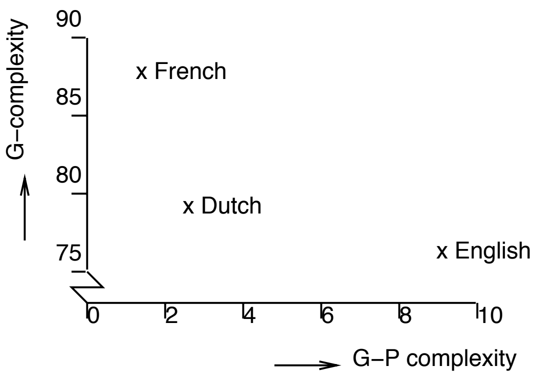

Figure 3. *Graphical display of the two-dimensional orthographic depth space, with 'x's marking the three corpora.*

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., it is enough to ensure that the learning algorithms are confronted with every regularity and irregularity of the writing system under consideration.

**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, with the Department of Psychology at Tilburg University, and with the Laboratoire de Psychologie Expérimentale 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.

## References

Carello, C., Turvey, M., Lukatela, G. (1992). "Can theories of word recognition remain stubbornly nonphonological?" 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'Année 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 Speech Synthesis Conference '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.

Gak, V. (1976). *L'orthographe du français.* Paris: S.E.L.A.F.

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.

Quinlan, J. (1986). "Induction of decision trees." *Machine Learning*, 1, 81–206.

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).

Van den Bosch, A., Daelemans, W. (to appear). "A distributed, yet symbolic model of text-to-speech processing." to appear in P. Broeder and J. Murre (Eds.), *Proceedings of the Workshop on Cognitive Models of Language Acquisition.*

Venezky, R. L. (1970). *The structure of English orthography.* The Hague: Mouton.