Journal of Memory and Language 72 (2014) 98–115

Contents lists available at ScienceDirect

Journal of Memory and Language

journal homepage: www.elsevier.com/locate/jml

When silent letters say more than a thousand words: An implementation and evaluation of CDP++ in French

Conrad Perry [a,*], Johannes C. Ziegler [b], Marco Zorzi [c,d]

[a] Faculty of Life and Social Sciences, Swinburne University of Technology, Australia
[b] Aix-Marseille Université, Centre National de la Recherche Scientifique, France
[c] Dipartimento di Psicologia Generale, Università di Padova, Italy
[d] IRCCS San Camillo Neurorehabilitation Hospital, Venice-Lido, Italy

ARTICLE INFO

Article history:
Received 13 November 2012
revision received 7 January 2014

Keywords:
Reading aloud
Connectionist modelling
Connectionist Dual Process Model
French

ABSTRACT

Cross-language comparisons can provide important constraints on our understanding of how people read aloud. French is an interesting case because it differs from most other writing systems in that it uses a large number of multi-letter vowel graphemes and consonants that are systematically silent (i.e., do not map to any lexical phonology; e.g., *trop*). Here, we developed a French version of the Connectionist Dual Process Model of Reading Aloud (CDP++) that can handle multisyllabic stimuli (up to three syllables) and has a large-scale lexicon of more than 100,000 words. We tested the model on extant data and an additional experiment examining the reading aloud of nonwords with potentially silent letters. The results from the extant data showed that the model was able to capture a number of important psycholinguistic effects in the literature and explained between 52% and 67% of the item-specific variance in two large databases. The results of the silent-letter experiment showed that, contrary to what would be predicted on the basis of lexical database statistics, people generally pronounce "silent" consonants in nonwords. We show that the French CDP++ model faithfully predicted this effect because it implements a *linear* mapping between orthography and phonology. These findings highlight the theoretical and practical significance of using computational models to help determine the processes and representations that underlie skilled reading.

Introduction

There are now a number of different models of reading aloud (e.g., Ans, Carbonnel, & Valdois, 1998; Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; Perry, Ziegler, & Zorzi, 2007; Perry, Ziegler, & Zorzi, 2010; Perry, Ziegler, & Zorzi, 2013; Plaut, McClelland, Seidenberg, & Patterson, 1996; Seidenberg & McClelland, 1989). Most of these models have been developed for English, which has been considered by some as an "outlier" orthography that has "shaped a contemporary reading science preoccupied with distinctly narrow Anglocentric research issues that have only limited significance for a universal science of reading" (Share, 2008, p. 584). While this statement seems overly harsh given the tremendous progress that has been made in the area of modelling word recognition (for a review, see Norris, 2013), it is important to test computational principles and implementations in other languages to see whether they are general or idiosyncratic to the language they have been developed for (for discussions, see Frost, 2012; Share, 2008; Ziegler & Goswami, 2006). In the context of connectionist learning models, it is crucial to investigate to what extent the same simple learning mechanism can produce quite complex and quite different

* Corresponding author. Address: Faculty of Life and Social Sciences (Psychology), Swinburne University of Technology, John Street, Hawthorn, Victoria 3122, Australia.

E-mail address: ConradPerry@gmail.com (C. Perry).

http://dx.doi.org/10.1016/j.jml.2014.01.003
0749-596X/

behavior that is found in different languages (e.g., Yang, McCandliss, Shu, & Zevin, 2009). In the present article, we will describe and test a French version of the latest *Connectionist Dual Process* (CDP) model. The basic architecture of the model appears in Fig. 1.

The basic architecture of the CDP++ model

The latest version of CDP, CDP++.parser (Perry et al., 2013) can be conceptually broken into two main parts, a lexical route and a sublexical route, which split-off from each other after the letter representations. The lexical route provides a simple look-up mechanism whereby words activate their whole-word entries in an orthographic lexicon. These entries then activate entries in a phonological lexicon, and finally phonemes in a phoneme output buffer. The sublexical route first selects graphemes from letters, a task performed by the graphemic parser. These graphemes are placed in a graphemic buffer, which is the input layer of the two-layer associative (TLA) network where linear relationships between graphemes and phonemes are learned. After this, phonological activation is generated. This activation is combined with the activation generated by the lexical route at the phoneme output buffer, and a pronunciation is derived. At present, both the graphemic parser and the TLA network learn from corpus data using the delta rule (Widrow & Hoff, 1960). This rule is formally equivalent to the Rescorla–Wagner rule (Sutton & Barto, 1981) and has been used to great success in other language studies (see e.g., Baayen, Milin, Durdevic, Hendrix, & Marelli, 2011).

One reason for the superior performance of recent versions of CDP, which differentiates it from the other models, is the nature of the sublexical route (see Zorzi, 2010). The use of a graphemic level of representation above individual letters is important both for theoretical and computational reasons (see Perry, Ziegler, Braun, & Zorzi, 2010; Perry et al., 2007; Perry et al., 2013). In terms of theoretical reasons, there are a number of neuropsychological disorders as well as experimental studies suggesting the importance of graphemes in reading and spelling (e.g., Caramazza & Miceli, 1990; Cotelli, Abutalebi, Zorzi, & Cappa, 2003; Howard & Best, 1996; Rey, Ziegler, & Jacobs, 2000; Tainturier & Rapp, 2004). In terms of computational reasons, adding a grapheme level improved the model's performance on word and most importantly on nonword reading. This can easily be seen by comparing the different versions of CDP. The first version of the model (Zorzi, Houghton, & Butterworth, 1998) did not use graphemes and its performance was much poorer than that of later versions.

If a graphemic representation is used in models of reading aloud, then it is important to specify how graphemes are selected from letters. The most recent version of CDP, CDP++.parser (Perry et al., 2013), showed how the mapping between letters and graphemes could be learnt using a parser that was constructed from a simple linear network with a memory. To get the parser to work, a database of words was first decomposed into their graphemes. After this, the words were presented to the model as strings of letters of a fixed length (5 letters for English), where each string contained a grapheme at the start of it. The model then learnt to choose the first grapheme in each of the strings, with the strings being presented to the model in the order in which the graphemes occurred in words. Thus, the input to the model was a string of letters and the output was a grapheme. The core idea behind the parser was that letters occur in a restricted attentional window, and this window moves from left-to-right across letter strings. This allows words of any length to be parsed, since all the

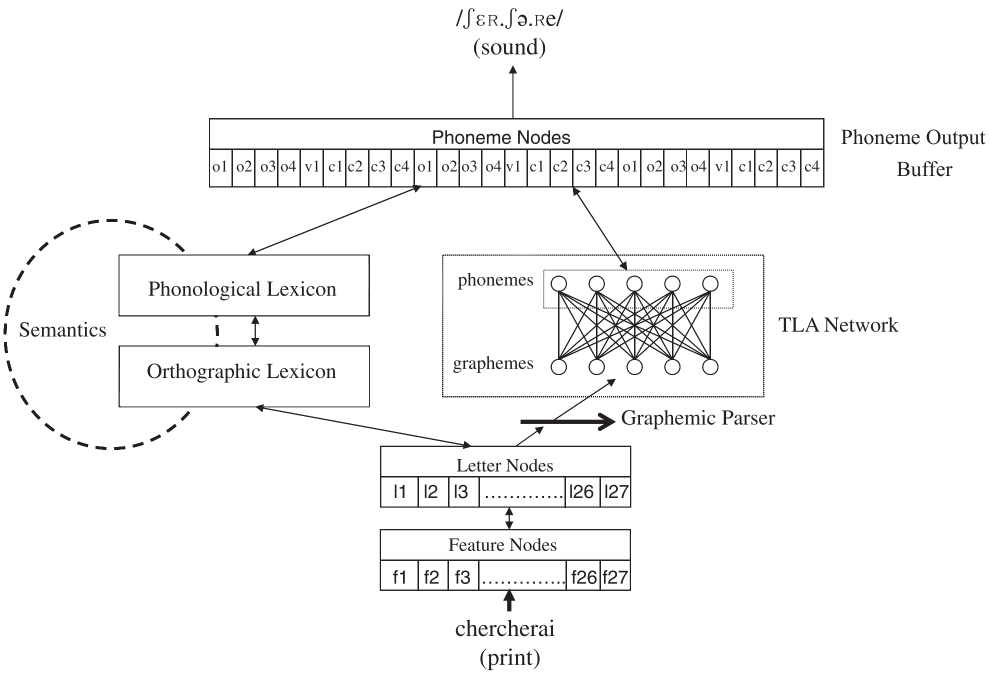

Fig. 1. The French CDP++ model.

model has to do is choose the grapheme at the start of the window at any given time step and this could theoretically occur indefinitely. For example, the letter strings used with the word *catcher* were *catch*, *atche*, *tcher*, and *er****, where the * represents no letter. From these letter strings, the model learnt to iteratively select the *c*, *a*, *tch*, and *er* graphemes.

Apart from just learning to select the graphemes, the model also learned whether the graphemes occurred in an onset, vowel, or coda category. This information is crucial to the model in *running* mode, where graphemes are selected online with no information from lexical phonology, because it allows the graphemes to be inserted into an orthographic template that is syllabically structured, such as that used with the TLA network used in CDP++ (see e.g., Perry, 2013; Prinzmetal, Treiman, & Rho, 1986; Taft, 2001, for evidence people may use syllabically structured orthography). Orthographic syllable boundaries can be found by categorizing graphemes, because if an onset grapheme occurs after a coda or vowel grapheme, it means that there must be a syllable break between them. For example, with the word *catcher*, if one knows that the –tch is an onset grapheme and that the –a is a vowel grapheme, then there is enough information to infer that the –tch grapheme should be placed into the onset of the syllable that occurs after the vowel, rather than in a coda position of the same syllable. With CDP++.parser, this information about what category graphemes occur in is given to the graphemic parser in learning mode, where the graphemes are aligned with lexical phonology as well as they can be, and the relationships between graphemes and the categories they should go in are then learnt. For the word *catcher*, the syllabic alignment and hence which category its graphemes should go in is simple because each grapheme in *c.a.tch.er* maps to exactly one phoneme, although obviously more complex cases exist. Without knowing whether –tch is a coda or an onset grapheme, some other method to place the grapheme into the template would need to be used. An evaluation of CDP++.parser showed that it performed better than CDP++ (Perry et al., 2010), a model which used a parser where graphemes are selected on the basis of simple rules.

The specificities of French orthography

French has a writing system with a fairly complex orthographic structure. It therefore makes an ideal cross-language test-case for whether the mapping between letters and graphemes can be learned. The French orthography differs from many others in a number of ways. One is that long sequences of vowel letters are common (e.g., *vieux*, *criaient*, *coeur*). This is potentially challenging for CDP++.parser because it is not simply given a rule by which to choose graphemes, but rather needs to learn how to choose them from letters.

A second peculiar aspect of French is that, like the letter –e in English which can occur in either a vowel or coda position (e.g., *be̲t*, *fate̲*), many graphemes constructed from vowel letters can have multiple roles, rather than just map to single vowel phonemes. For example, the grapheme –ou maps to the semi-vowel /w/ in some circumstances (e.g., *joua* [played] – /jw̲a/) and a vowel in others (e.g., *brouillent* [scramble] – /bʀu̲j/). This pattern is also challenging for CDP++.parser, because it means that the parser must learn that the same grapheme needs to be categorized as an onset (which is where the semi-vowel needs to be placed in the sublexical network of CDP++) or a vowel, depending on the letter context in which it occurs.

A third interesting pattern in the French orthography that makes it different from many other languages is that large numbers of letters are silent – that is, they do not appear to map to any phonology. This pattern is systematic, rather than something that occurs in an ad hoc manner. These silent letters do not appear to be restricted only to the end of words, but they can occur at the end of syllables. For example, the –p in *baptiser* [baptize] (/batize/) is silent (e.g., Chetail & Content, 2013). To confirm this, we performed an analysis of just the first syllable of words in the database described below. For the sake of simplicity, we removed bodies with a final silent –ent morpheme (e.g., *ils regardent* [they look]) as well as words where a final –es was silent after following another /s/ (e.g., *caisses* [boxes]). The results showed that there were 1500 orthographic bodies that exist in French that do not end in the letter –e and do not just contain vowel letters. Of these, 369 always have the final consonant pronounced (e.g., –el in *tel* [such]), 1019 always have at least one silent final consonant (e.g., -ottes in *bottes* [boots]), and 112 have at least some level of ambiguity, where the letters may be pronounced or may be silent, and these may differ in terms of how many of the letters are silent when there is more than one letter. For example, the word *est* [is] uses no coda phonemes but *est* [east] uses two, the body –omp uses one coda phoneme in the first syllable of *somptueux* [sumptuous] but none in *compter* [to count], and the body –ord uses one coda phoneme in *bord* [edge] and two in *fjord* [fjord]. With a model like CDP++, some hypothesis about how these letters are dealt with needs to be made and its behavior investigated to see whether it makes interesting and testable predictions about how human participants deal with silent letters in nonwords.

The graphemic parser of the French CDP++ model

The graphemic parser of CDP++ is constructed of a simple two-layer network that takes a string of letters as input and tries to predict what the most likely grapheme is at the start of the string as well as its category (i.e., onset, vowel, coda). The input also has a memory of previous graphemes that it has chosen, and the categories are determined by having 3 slots (onset, vowel, and coda) in the output where the graphemes are duplicated. When the model is run, the 'winning' grapheme is the one that is activated the most across the 3 slots. The current version is parameterized identically to the English version apart from the number of letters (37 in the French alphabet plus 1 for the blank letter) and graphemes (111 were used, including single-letter graphemes). It uses 5 slots for the letter input, where any potential letter in the French alphabet can go, and there are 3 time-steps for the memory of the previous graphemes. This means the input into the network is simply a vector constructed of letters (38 letters × 5 letter spots) and previous graphemes (111 graphemes × 3 slots × 3 time steps) and the output is a vector of graphemes (111 graphemes × 3 slots). The network appears in Fig. 2.

The French parsing mechanism was developed based on the same principles as CDP++.parser. This involves constructing a training database in which words are segmented into graphemes (see Perry et al., 2010, for a discussion of issues to do with grapheme selection, and Appendix A for specifics of how the database of over 100,000 words was developed, more specific details about the model, as well as an evaluation of the error rates). An important language specific issue is that that some solution is needed to represent semi-vowels and silent letters so that the graphemes in words can be coded.

In terms of semi-vowels, based on the idea of using one-grapheme–one-phoneme mappings where possible, we split the vowel letters that contained them so that two graphemes are represented, one where the letters map to the semi-vowel and one where the letters map to the vowel, and the semi-vowel was learnt in the onset category. In most cases, this means the first letter of the vowel sequence is used as the semi-vowel, but there are infrequent cases where more than one letter is used. For example, with the –oua in *joua* [played], -ou and –a graphemes were used, with the –ou being used in the onset position (mapping to /w/) and the –a as a vowel. This means that the parser has to be able to categorize the –ou graphemes as occurring in an onset or vowel position depending on the letter context in which they occur.

In terms of silent letters, the approach used to code the graphemes was to use the simplest set of graphemes that typically map to phonology and then get the sublexical two-layer associative (TLA) network of CDP++ to *try* to learn that certain graphemes do not produce phonology (see below for further details). The simplest set of graphemes for each word with silent letters was found by selecting the shortest graphemes that can occur in onsets and vowels and by using them in coda-final positions. So, for example, the graphemes in *arrhes* [deposit] /aʀ/ were specified as –a, -rr, -h, -e, -s (where the –e is a coda and not vowel grapheme) rather than –a and –rrhes, the latter of which has the same number of graphemes and phonemes. The –h, -e, and –s were used as they often map to single phonemes when not at the end of words. This means that, just like in English, the –e can function in both a vowel and a coda position. That is, in some words, it acts as a vowel (e.g., *vert* [green] – /vɛʀ/) whilst in others it acts as a coda to mark that the final consonant should be pronounced (e.g., *visage* [face] – /vizaʒ/).

Simulations with a full-blown CDP++ model of multisyllabic reading in French

The first thing we did in the construction of the new model was to train and test a new parser. The parser was identical to that in Perry et al. (2013) except that it used the set of French letters and graphemes and not the English ones and was trained on the new database. Since it displayed excellent results, we integrated it into CDP++ to provide a full blown CDP++ model that is capable of simulating French reading aloud (the model is available online). To do this, a French version of CDP++.parser was constructed that was identical to the English one except that it used the French database and it did not have stress nodes since French does not have word-level stress (e.g., Dupoux, Pallier, Sebastián, & Mehler, 1997). See Appendix A for further details.

Simulation 1: Quantitative performance of the model

To evaluate the model, all words in its lexicon were first run with the lexical route turned off (i.e., only sublexical phonology was generated). Note that, like the English model, one would not expect perfect performance. This is

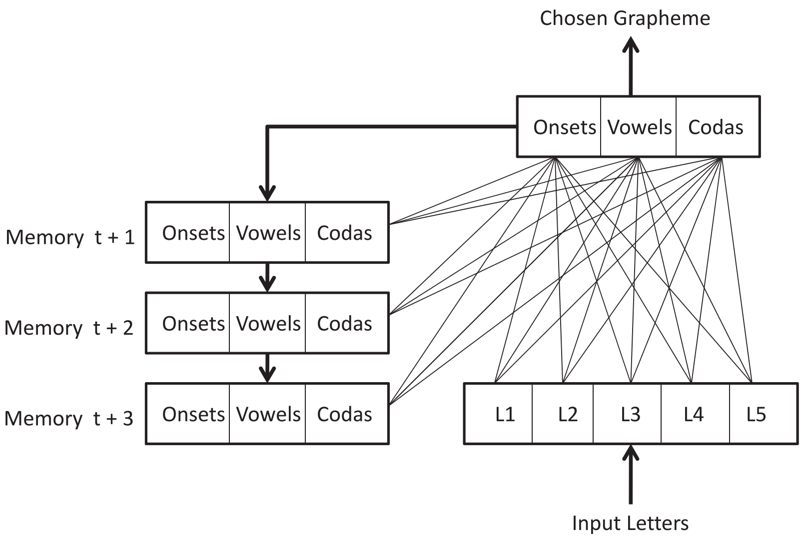

Fig. 2. The graphemic parser. *Note: t* = time; *L* = Letter.

Table 1
Percentage of variance explained (r^2) by CDP++ RTs plus onset coding and a number of regression equations on the Courrieu, Brand-D'abrescia, Peereman, Spieler, and Rey (2011) and Ferrand et al. (2011) databases.

Data set	CDP++	Regression		
	Onsets+ CDP++ RTs	Onsets+ Frequency+ Length+ Levenshtein distance	Onsets+ Frequency	Onsets
Courrier et al.	66.5	68.2	62.5	55.5
Ferrand et al.	52.2	52.7	50.1	44.6

because there are unpredictable irregularities in French, as in most other orthographies, and because the sublexical route of CDP++.parser only allows linear relationships to be learnt. This means that it can never learn all of these irregularities, unlike networks that do allow nonlinear relationships to be learnt (e.g., Plaut et al., 1996). The results of the model showed that it produced the correct answer 83.1% of the time.

The next aspect of the model we examined was its quantitative performance.[1] This was done in the spirit of Spieler and Balota (1997), who argued that a good test of a model is that the values it produces at the individual item level should correlate as well as a simple regression using factors known to be important in reading. To examine this performance, two large French databases (Courrieu, Brand-D'abrescia, Peereman, Spieler, & Rey, 2011; Ferrand et al., 2011) were used. The database of Ferrand et al. has naming latencies for 1482 monosyllabic words and the database of Courrieu et al. has naming latencies for 615 disyllabic words. For the model here, we compared the performance with a number of regression equations that included the onset coding scheme used in Courrieu et al., orthographic word frequency, orthographic word length (i.e., number of letters), and orthographic word neighborhood calculated with Levenshtein distance scores based on the 20 closest neighbors (see Yarkoni, Balota, & Yap, 2008 – this is a measure of how many other words are visually similar to a target word). Like Perry et al. (2013), for these and the other simulations, we used a feedforward model with only items with an identical orthography or phonology as the word being presented in the lexicon. This meant that only the word presented and words that were homophonous with it were in the lexicon, and nonwords were run with no items in the lexicon. The only exceptions to this were made for tasks that required feedback to capture the effect, which in this case was only the pseudohomophone effect, where a full lexicon was used. This was done for practical reasons because, for most aspects of the model, the performance of the model without feedback is almost identical (see Perry et al., 2007; Zorzi, 2010), and the simulations based on a full lexicon are exceptionally slow. In terms of further data processing, words with naming latencies more than 2.5 *SD*s away from their means were removed from the analysis, with separate means and *SD*s being calculated for each cell from the designs used in the experiments below (e.g., if the manipulation included, for example, words and nonwords, the means and *SD*s would be calculated for words and nonwords separately). The results of the databases are displayed in Table 1. Further information about the number of errors, outliers that were removed, and human means from the experiments appears in the Supplementary materials.

As can be seen, the model correlated almost as well as a regression equation with four predictors and higher than just onsets and frequency alone. This suggests that, like the English model, the model is able to pass a strong quantitative test.[2] Interestingly, there was a difference between the English (see e.g., Perry et al., 2010) and the French data in that there was in fact less variance to capture that was not due to variation in the onset characteristics of words in French.[3]

Simulation 2: Effects of regularity/consistency

In addition to the quantitative evaluation, there are a number of factorial benchmark experiments in French that are also worth testing the model on. One of these is the regularity/consistency effect, which is well known in the English literature (e.g., Jared, 2002). The basic idea is to examine the extent to which spelling–sound correspondences in words that are different to those that could be determined by a set of rules (regularity) and the extent that the same spellings map to different sounds (consistency) affects people's reading performance. Results from studies examining this in French (Content, 1991; Ziegler, Perry, &

[1] It would of course have been interesting to compare the results of CDP++ with Ans et al. (1998). Unfortunately, simulations from the Ans et al. model were not made available to us.

[2] This is actually very good because a regression equation with four variables fits the actual data with 5 free parameters whilst CDP++ was not tuned to capture the maximum amount of variance in just this particular database.

[3] Apart from comparisons with regression equations, we also examined the maximum amount of reproducible variance using the method of Courrieu, Brand-D'abrescia, Peereman, Spieler, and Rey (2011). Using this method, a value of $r = .92$ was found by comparing groups of 50 participants that were in the Courrieu et al. database (there were 100 participants in total in the study and this therefore represents the largest group size that could be compared). This is clearly higher than the prediction of the model. However, it is not clear to what extent the variance that is not explained by the model could ever be captured. For example, the variance could be due to the onset coding scheme being simpler than the real phonetic properties of words, error in database frequency counts, factors that are not implemented in the model that might potentially be of interest, such as age of acquisition, and factors not implemented in the model and of little interest, such as idiosyncratic properties associated with individual words. Given this, it is not clear to what extent the model is really underperforming compared to the maximum amount of reproducible variance. Nevertheless, it is clear that it may be possible for another model to perform quantitatively better than the model which is proposed here.

Coltheart, 2003) have found that if a word has a subsyllabic spelling–sound correspondence that is unpredictable out of word context, people are slower at reading it aloud than if it does not, and this occurs with both high and low frequency words. To simulate these effects, the items in Content and Ziegler et al.'s experiments were run through the model using the same set of parameters that were used with the large databases. The results appear in Fig. 3.

The model showed essentially the same pattern of results across three experiments. With the Ziegler et al. (2003) items, the model showed main effects of Regularity, $F(1,51) = 10.75$, $p < .005$, and Frequency, $F(1,51) = 10.48$, $p < .005$, but no interaction, $F(1,51) = 2.60$, $p = .11$ (High Frequency Irregular: 89.4; High Frequency Regular: 77.7; Low Frequency Irregular: 93.3; Low Frequency Regular: 89.3; Variance Explained: 16.3%). Note that the trend towards there being a significant interaction was actually caused by the high-frequency items showing a larger effect than the low-frequency ones. With the items from the first experiment of Content (1991), the results were very similar to the Ziegler et al. items, with main effects of Frequency, $F(1,64) = 22.87$, $p < .001$, and Regularity, $F(1,64) = 21.41$, $p < .001$, but no interaction, $F(1,64) = 1.23$, $p = .27$ (High Frequency Irregular: 98.3; High Frequency Regular: 87.9; Low Frequency Irregular: 114.9; Low Frequency Regular: 98.2; Variance Explained: 11.0%). With the items from the second experiment of Content, the model again showed the same pattern, with main effects of Regularity, $F(1,116) = 20.97$, $p < .001$, and Frequency, $F(1,116) = 64.1$, $p < .001$, but no interaction, $F < 1$ (High Frequency Irregular: 100.7; High Frequency Regular: 90.4; Low Frequency Irregular: 117.5; Low Frequency Regular: 108.4; Variance Explained: 32.6%). Given the results of the three experiments, the predictions of the model converge with the empirical data to suggest that, in French, there are main effects of regularity and frequency, but no regularity by frequency interaction. The quantitative performance of the model was also reasonable.

Whilst it appears that the model can produce results that are similar to those of people, it is also useful to examine what makes words difficult in French since some forms of inconsistency that exist may be more common than in other languages. One type is where a grapheme can map to more than one phoneme. This is very common in English where, for example, with a word like *chef*, an effect of inconsistency occurs because the –ch grapheme typically maps to /tʃ/ and this causes competition with the correct phoneme /ʃ/ which is generated lexically. This then slows processing down. This type of inconsistency is also common in French. For example, the vowel in *hall* [hall] is /o/ but most words with an –a use /a/ (e.g., *mars* [March, the month]). Another type of inconsistency is to do with silent letters. As noted above, words with inconsistent bodies of this type might be harder to process, as has typically been assumed in studies examining the regularity effect in French. Thus, it is interesting to examine how words with silent phonemes that use bodies that are not typically silent (e.g., *broc* [large jug]) are processed and vice versa (e.g., *mars*). To do this, we examined the activation produced by a number of words with these properties that the model produced slower than average reaction times on. These appear in Fig. 4.

The first three words in the figure (*hall, broc, mars*) are taken from Ziegler et al. (2003). *Hall* represents a word

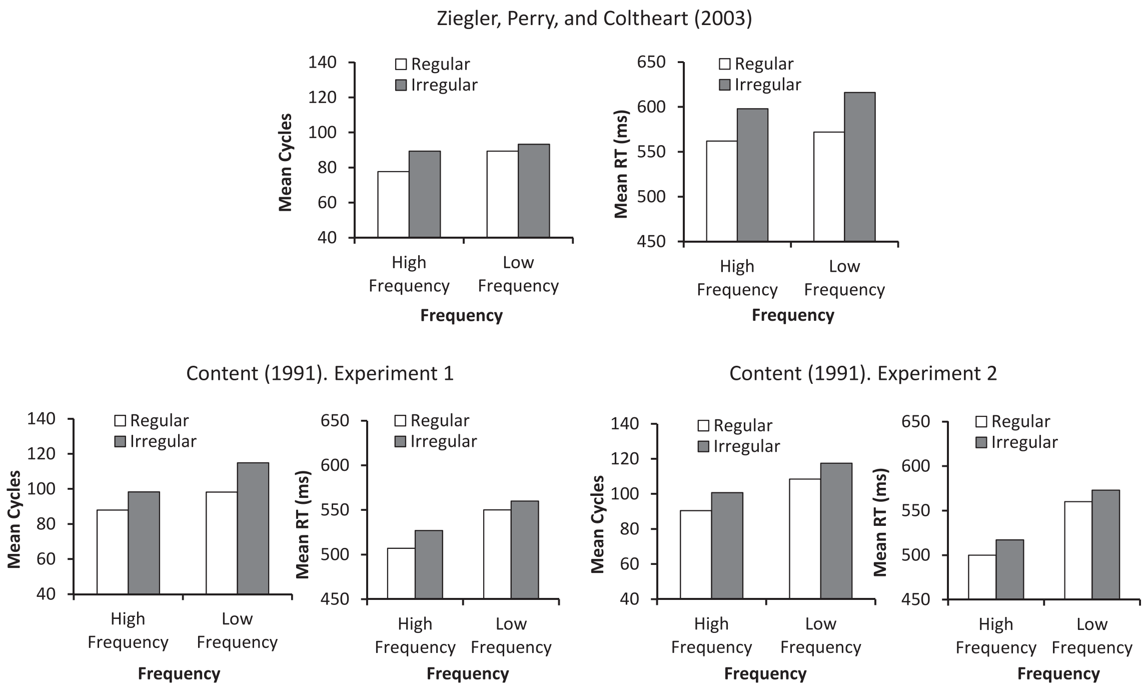

Fig. 3. Performance of CDP++ (mean latency in number of cycles) and skilled readers (mean RTs in milliseconds) on the high and low frequency regular and irregular words used in Ziegler et al. (2003) and the two experiments of Content (1991). *Note:* ms = milliseconds.

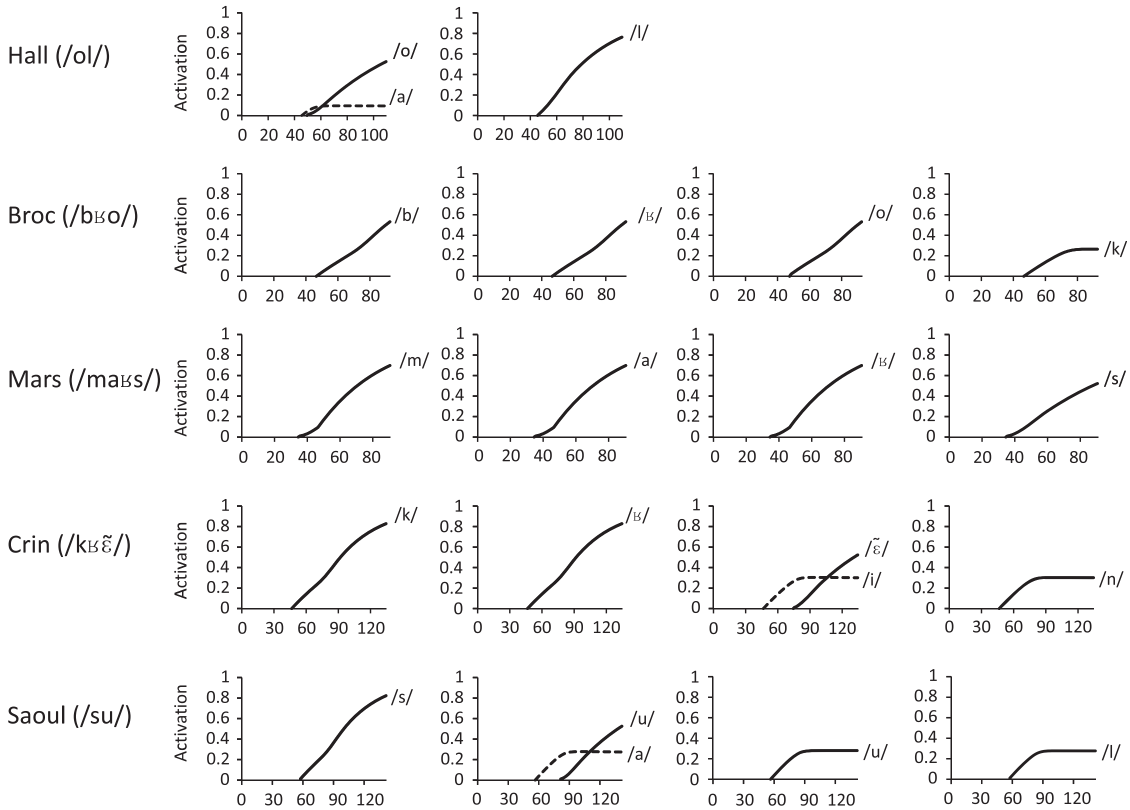

Fig. 4. Activation of individual phonemes in a number of different words over time (cycles).

with an inconsistent vowel, *broc* a word with a silent final consonant that would typically be produced, and *mars* a word with a final consonant that would typically be silent (i.e., -s). As can be seen with *hall*, there is competition between two different vowels, the incorrect one which is produced by the sublexical route and the correct one produced by the lexical route. This competition causes the processing of this vowel to slow down compared to the only other phoneme (/l/) and hence the word is produced more slowly than if it had a vowel that was entirely consistent. In terms of the word *broc*, as can be seen, the final –c is activated by the TLA network. This causes a slow-down in processing because the lexical route must inhibit the phoneme so that its activation is no longer increasing. Once this happens, the model knows not to include the phoneme in its final pronunciation and, since all other phonemes are above the phoneme naming activation criterion (i.e., the activation criterion at which phonemes are included in the final pronunciation), it allows the model to terminate processing. The third word *mars* is slowed down with the /s/ phoneme because, unlike words that use a letter that consistently activates a phoneme, the TLA network produces less activation for –s because -s is typically silent in the 2nd coda position. Thus, rather than having a full contribution of both lexical and sublexical activation, a full contribution is produced only by the lexical route. These three words show that the actual reason a phoneme may be slowed down within words can differ and there are two main ways these effects are caused. One is due to learning where the strength of the connection between a grapheme and the correct phoneme is reduced when other words using the same grapheme map to a different phoneme, or, in the case of silent letters, when the grapheme maps to nothing. This makes the strength of activation of the correct phoneme weaker. The other is because the network can only learn linear relationships and simply produces the incorrect phonemes with some words. The activation from the incorrect phoneme then causes interference with the correct phoneme which can be generated lexically and, to a lesser extent, sublexically (i.e., both the incorrect and correct phonemes may be activated sublexically, but the level of activation usually differs).

Apart from just words that were slow to process but were still used in the analysis, we also examined the 5 unique outliers that occurred across the 248 words in the above studies (*noce*, *crin*, *gaie*, *sonnerie*, *blairer*) as well as the word *saoul* which was close to being an outlier and also showed a pattern of interest. Three main patterns could be identified in the data. One was when an incorrect vowel was produced that was later lexically corrected (*crin*, *gaie*, *saoul*, *blairer*) and a second similar pattern was when incorrect activation was generated from a consonant (*noce*, which activated both /k/ and /s/ in the same coda position). The third pattern occurred when the parser created a pattern of graphemes that caused phonology to be activated in places that were not aligned with lexical phonology. In particular, with the word *crin*, the graphemes used were c.r.i.n vs. c.r.in, and *saoul* was s.a.ou.l vs. s.aou.l (note that the first parsing appears reasonable, simply not the one needed for *saoul*). In this case, placing graphemes in the wrong places caused problems because it caused the incorrect phonology to be activated which then needed to be lexically corrected. As can be seen in Fig. 4, with *crin* and *saoul*, the actual dynamics causing words to be outliers is similar to words that are read-aloud more slowly than average but do not cause outliers, except that outliers almost inevitably have large amounts of incorrect phonology competing with the correct phonology (note that with *saoul*, the final /u/ and /l/ phonemes are incorrectly produced in the second syllable). Thus, like is found in real data (see e.g., Andrews & Heathcote, 2001), the model can generate reasonably long tails. This is also true of even the mean item values in the studies that were simulated. In Content's (1991) first experiment, for example, the fastest low frequency irregular word (*opus*) had a mean reaction time of 464 ms and the slowest (*gille*) had a mean reaction time of 800 ms. If part of the time it takes to respond is reasonably constant across words of similar length and frequency (e.g., articulation, early visual processing), this suggests that it is likely that the variance in the amount of time it takes the other processes to complete may be quite large.

Simulation 3: Effects of syllable number

Another interesting effect that is particularity relevant for models of multisyllabic reading is the effect of syllable number. Ferrand (2000) examined this effect in French and found that, after controlling for letter length, the more syllables a low frequency word or nonword has, the longer it takes to read it aloud. Unlike the regularity effect, this effect does not occur with high-frequency words. In terms of the word data, the model showed a very similar pattern, with a significant main effect of Frequency, $F(1,73) = 46.1$, $p < .001$, a main effect of Syllable Number that was not significant, $F(1,73) = 2.23$, $p = .14$, and a Frequency by Syllable Number interaction that was significant, $F(1,73) = 4.60$, $p < .05$. Two *t*-tests showed that the Syllable Number effect was significant with the low, $t(36) = 2.22$, $p < .05$, but not the high frequency words, $t(38) = 1.00$, $p = .32$ (High Frequency 2 syllables: 102.8; High Frequency 3 syllables: 101.8; Low Frequency 2 syllables: 110.2; Low Frequency 3 syllables: 116.0). Unlike the human data, the model did not show a Syllable Number effect with nonwords, $t < 1$ (2 syllables: 155.6; 3 syllables: 153.9). The model also made 3 errors on the nonwords (2.5%), where we considered errors to be nonword pronunciations that used a grapheme–phoneme correspondence that does not exist in real words, excluding very atypical spelling–sound correspondences found in loan words.

The results of the simulations are somewhat surprising in that the low frequency words showed a syllable number effect but the nonwords did not, which appears inconsistent with the claim that the syllable number effect might occur because of differences in the processing of the sublexical route (Ferrand, 2000). In the English model, for example, one reason a syllable number effect is found is because the correct segmentations are relatively difficult to generate, and thus syllable breaks cause ambiguity which then causes slower processing. In French, however, the syllable breaks are simpler and easier to find (as seen by the accuracy of the parser), and thus one would not expect such a strong effect. An alternative possibility for the result is that the low frequency items are confounded on lexical characteristics, and thus there is no sublexical effect in both cases. This can be seen by running the model without sublexical phonology. When this is done, a significant difference between the low frequency words with two and three syllables is still found (126.1 vs. 135.5 cycles), $t(36) = 2.22$, $p < .05$, suggesting that the effect produced by the model has largely a lexical, not sublexical, origin. Investigating the words showed why the model produced this result. At least with the database we used, the log frequency of the low frequency words was confounded, with the two syllable words having a higher mean log frequency than the three syllable words ($t(36) = 2.54$, $p < .05$, two syllable log frequency: 2.70; three syllable log frequency: 2.18), and the model is very sensitive to this factor.

Assuming that the low-frequency stimuli are confounded on lexical characteristics, and thus that the real results produced by the model would be closer to a null effect if the low-frequency stimuli were better matched, this leaves the syllable number effect unexplained by the model. However, this effect may be out of the scope of the model, since an alternative explanation is that it may be due to an articulatory planning effect, which occurs at a level whose properties are not specified in the model. In this case, the slowdown in the three compared to two syllable nonwords may have been caused by differences in the speed at which articulatory encoding occurs. Support for this comes from the study of Meyer, Roelofs, and Levelt (2003), who found that, in some conditions, syllable number effects emerge with pictures. They suggested that this occurs because turning a string of phonemes into an articulatory code does not happen in parallel, but rather segments are incrementally parsed into syllables. This means that the more syllables a word has, the longer this process takes. This explanation has recently been used to explain data found in a reading aloud task in Italian by Sulpizio, Arduino, Paizi, and Burani (2013). They found that participants read aloud 4 syllable words around 100 ms more slowly than 3 syllable words. These authors suggested that taking articulatory planning into account was necessary to explain their results. Part of their explanation of articulatory planning included the idea that it may differ based on how well a word is learnt, with high frequency words potentially benefitting from the use of whole-word articulatory units. This would reduce the size of the syllable number effect they would display compared to low frequency words and particularly nonwords that could not benefit from this at all. If this explanation is correct, then it would predict the pattern of results found, but, for CDP++ to capture the data, it would require the implementation of an articulatory level that currently does not exist in the model.

Apart from the Italian data, further evidence that the syllable number effect might be articulatory, or at least not entirely from phonology generated sublexically, comes from a comparison with the regularity effect. With the regularity effect, all French studies examining it have reported a similar sized effect with both high and low frequency words. With the syllable number effect of Ferrand (2000), however, no effect was found with high frequency words. If the locus of the effect is the same, where difficulties in generating sublexical phonology across syllables causes a slowdown in naming times, then it is not clear why the pattern of effects should differ. This suggests that an explanation like Ferrand (2000) gave for his data, where it was assumed that the phonology of high frequency words could be generated lexically so quickly such that no sublexical effects would be found, is unlikely to be correct. Thus an alternative explanation, such as that suggested by Sulpizio et al. (2013), may be better.

Simulation 4: Pseudohomophone and neighborhood effects

A final set of results of interest are those of Grainger, Spinelli, and Ferrand (2000). They examined two different effects in the same experiment. One was the pseudohomophone effect (i.e., nonwords that sound like words, e.g., *avryl*, which is a pseudohomophone of *avril* [April]) and one was the effect of orthographic neighborhood measured in the traditional way (see Coltheart, Davelaar, Jonasson, & Besner, 1977). The results they found showed that people named pseudohomophones faster than nonword controls. They also found that words and nonwords in a high orthographic neighborhood were faster to read aloud than words and nonwords in a low orthographic neighborhood. Using the same items, the model produced a pseudohomophone effect, $F(1,201) = 4.00$, $p < .05$ (Pseudohomophone: 125.8; Nonword Control: 128.3), but did not show a neighborhood effect, $F < 1$ (Low Neighborhood Nonwords: 126.3; High Neighborhood Nonwords: 128.4; Low Neighborhood Words: 91.0; High Neighborhood Words: 91.6). The model also had an error rate of 8.3% on the nonwords and pseudohomophones. Two (.83%) of these errors were legitimate possible alternative readings of pseudohomophones, 5 (2.9%) were a number of implausible word captures where a short nonword was lexically captured by a long one (e.g., *fabricant* [a fabricator] for *fabri*), and 4 (1.6%) occurred because the letter –j was in a graphemic position where little was learned (i.e., a dead-node (see Perry et al., 2010)). The lexical capture results suggests that the level of feedback in the system might potentially be set more accurately and also that other lexical routes that do not suffer from this property might be worth investigating with the model in the future.

Experiment

Given that nonword reading is a crucial benchmark for computational models of reading aloud (see Perry et al., 2007), it is important to further examine the ability of the model to generate accurate pronunciations to nonwords. This is important because, as noted in Perry et al. (2010), parameter fiddling makes it possible to improve the model's fit with respect to response latencies, but subtle parameter changes have a very limited effect on the actual pronunciations that are generated for nonwords. To examine this, we looked at the way people process letters that are typically silent when they occur at the end of words, thereby exploiting the peculiar characteristic of the French orthography described in the *Introduction*. The basic idea of this experiment was to test whether letters that are typically silent at the end of words are actually silent in nonwords.

The reason this is an interesting modelling issue is because learning whether letters are likely to be silent allows two different types of models to be compared: One where silent consonants are attached to vowels to form graphemes, and one where the consonants are separate. Thus, for example, if consonants are attached to vowels, then the graphemes in a word like *trop* [very] (which has a silent –p) would be *t.r.op*, whereas if they are separate, the graphemes would be *t.r.o.p*.[4] With the French CDP++, we implemented the latter of these schemes.

In terms of predictions, if consonants are attached to vowels, then, even with a linear network, it should be possible to learn that a given grapheme with consonants on the end produces only a vowel phoneme. In this case, there would only be ambiguity in the selection of graphemes, because the graphemes at the end of words that overlap at the letter level would be processed orthogonally at the graphemic level (e.g., "Produce no phonology for the consonant if there is an –op grapheme that ends a word, but produce /p/ if there is a –p grapheme"). Alternatively, if the consonants are represented separately, then this mapping will not be learnt perfectly because CDP++ uses a linear network to learn relationships between graphemes and phonemes and the –p would be overlapped by words with –pe on the end (e.g., *pape* [pope] – which we assume uses the graphemes *p.a.p.e*). The fact that the relationships between spelling and sound is nonlinear for a letter sequence like –pe might not be obvious from a rule based account, which has a very simple solution like "pronounce /p/ when there is an –e following the -p but not if there is no –e". In this case, the relationship might appear linear if a simple additive solution is used and each grapheme contributed equally to activating the /p/ phoneme (i.e., $p + e > e$ and $p + e > p$, meaning that the activation generated by –p.e together could be higher than –p or –e alone, hence potentially allowing /p/ to be produced for –p.e. but not –p or –e based on differences in activation). However, using this solution would mean that –e would need to become associated with and hence produce activation for the phoneme /p/, and indeed every other phoneme where this pattern appears (e.g., -b.e→/b/, -d.e→/d/, etc.), thus predicting that every time –e appears, many phonemes would be partially activated. A logical alternative is that –p generally produces most of the activation for /p/ except when it is inhibited by other graphemes, such as those in the vowel and the onset. If this is what occurs, then the model would

[4] We are talking about potential graphemes here, since whilst there is reasonable evidence that graphemes of some kind exist (see Perry et al. (2013), for a discussion), and whilst some people have tried to specify the set of possible graphemes (e.g., Venezky (1970)), the exact graphemes that people use in particular contexts and why is currently not perfectly specified.

need to learn not to produce /p/ based on quite complex relationships between –p and other letters, such as the vowels. Given that CDP++ cannot learn complex relationships, it could only do this in cases where this information is easily obtainable because it exists and is relatively frequent, and this differentiates it from other PDP style models that allow highly nonlinear mappings to be learnt (e.g., Ans et al., 1998; Plaut et al., 1996).

In this Experiment, we examined the extent that people generate phonology with potentially silent letters using two types of nonwords: one where the body exists in words and one where it does not. This manipulation was used because all versions of CDP++ allow some effects of letter context to be learnt. Thus, because the phonology CDP++ produces is affected by units larger than single graphemes, it may be more likely to produce phonology similar to words with nonwords that share bodies with other words than with nonwords with no shared bodies.

Method

Participants

Thirty-two undergraduate psychology students from the Université de Provence participated in the task. They came from the same experiment as Ziegler et al. (2003).

Stimuli

Seventy monosyllabic nonwords were used, although only 28 of them constituted the critical stimuli. Of these, all were monosyllabic, with 17 using an extant orthographic body and 11 a non-extant orthographic body. With the nonwords with extant bodies, all shared their bodies with words where the final consonant was silent in the first syllable of all words according to the Lexique 3.62 database (New, Pallier, Ferrand, & Matos, 2001). The only exceptions made to this were when the bodies existed in foreign loan words (almost all of which were English). Statistics for the nonwords appear in Appendix B.

Procedure

The nonwords were presented directly after completing the word reading task described in Ziegler et al. (2003). Before that task, the experimental procedure was explained to each participant. This included telling the participant what a nonword is and that they should read aloud the nonwords as quickly and as accurately as possible. In terms of the individual items, for each trial, a fixation point was presented for 500 ms which was then followed by the nonword in lower case letters for 500 ms. All presentation was done using PsyScope (Version 1.1; Cohen, MacWhinney, Flatt, & Provost, 2003). Responses were also recorded with a digital tape recorder for offline scoring.

Results

All pronunciations were first examined by a native French speaker who, where possible, tried to code them by hand. A small number of responses (.67%) were lost due to problems such as the subject coughing, problems with the microphone, etc. All responses that consisted of phonemes that did not occur based on typical French spelling–sound relationships were also removed from the analysis (5.92%). This included responses where the participant said what appeared to be part of a response and then stopped or revised themselves.

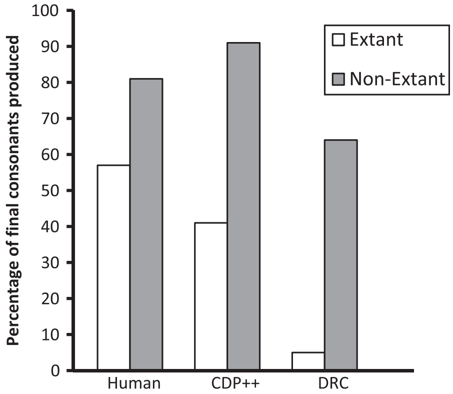

Fig. 5. Percentage of final consonants produced in the human data, CDP++, and the DRC on nonwords with extant and non-extant bodies.

The results showed that people often gave pronunciations to letters in nonwords that are typically silent in words – indeed, they pronounced "silent" letters in nonwords most of the time (Extant body: 57.8%; Non-extant body: 81.3%). The results of CDP++ showed that it produced very similar results to people, although somewhat under-predicted the number of non-silent consonants given to nonwords with extant bodies (Extant body: 41.2%; Non-extant body: 90.9%) (see Fig. 5). Apart from CDP++, we also examined the predictions of the French DRC (Ziegler et al., 2003). That model also predicted a difference between extant and non-extant bodies, although it did not produce consonants with nearly enough of the nonwords with extant bodies (Extant body: 5.8%; Non-extant: 63.6%). Quantitatively, both CDP++ and DRC had only very weak correlations with the reaction time data, with CDP++ explaining 1.3% of the variance and DRC 4.8%.

Discussion

The results of the experiment are important because the theoretical debate about reading aloud is largely focused on effects that occur with words and that reflect the statistical features of lexical databases. Here, however, we examined a pattern that does *not* obviously exist in word data. That is, we found that nonword pronunciations differed from what would be expected on the basis of analogies to real words or lexical databases. This means that we need a theory to explain why people display non-optimal reading performance in the sense of deviating from what would be expected on the basis of lexical statistics. Based on an a priori property of CDP++ – that it uses a linear network to learn the relationships between spelling and sound, and hence cannot learn nonlinear patterns – such a pattern was predicted and found (the issue of whether CDP++ is really linear will be addressed in the Section 'General discussion'). Thus, the model predicts an effect that diverges from obvious database statistics based on the underlying computations that are implemented to simulate reading aloud. The results of the DRC were also interesting as the set of GPC rules implemented by Ziegler et al. (2003) in the French DRC was chosen by hand to try and get the model's pronunciations as close as possible to the lexical data, thereby following the typical DRC modelling strategy. As can be seen, such rules cannot account well for the nonword data; the discrepancy between the DRC and human data in terms of pronouncing the "silent" letters was particularly striking for the extant body condition (5.8% vs. 57.8% for the DRC and the human data, respectively). There may be strategies that could potentially get the model closer to the real data, such as by choosing the rules of the model probabilistically or by using algorithms that learn the rules. However, when the rules of the DRC have been learned in English, the results have been substantially worse than the hand-picked set (Pritchard, 2013). Thus, whether such strategies would work in French without lowering the performance of the model remains unclear.

It is worthwhile noting that these results were not simulated by CDP++ as a mixing process between lexical and sublexical phonology, where the partial lexical activation of words meaningfully changes the nonword pronunciations. If this occurred, the results might be potentially described as occurring due to a sublexical plus lexical analogy mechanism. Rather, the results are driven by learning in the TLA network, where the extent that coda graphemes activate phonology and the extent that other graphemes inhibit phonology differs. Thus, running the model with no items in the lexicon gives almost identical results to running the model with a full lexicon.

Given that our experiment included only nonwords, one might wonder whether including word stimuli would change the present results towards a smaller proportion of "silent" letters being pronounced. In the CDP model, nonwords generally produce weaker activation of the output phonemes than words, because they do not get support from the lexical pathway. Because of this, we have argued (e.g., Perry et al., 2007) that people are likely to use a lower phoneme naming activation criterion (i.e., the criterion which phonemes must go over so that they can be entered into the final pronunciation) with stimuli sets where only nonwords are used compared to ones where words are also used (see also Kinoshita & Lupker, 2003). This means that if the nonwords were embedded in a context which forced this threshold to be higher, the average number of silent phonemes should increase. To investigate this issue in a more formal way, we ran all of the items used in the experiment with only the sublexical route for 250 cycles, ignoring the stopping criterion. To test the model, we also created a set of control nonwords that were identical to the experimental nonwords with the exception that they had an –e on the end (e.g., the control for *lont* would be *lonte*). Note that the latter type of nonwords are almost always produced with the final consonant (similar nonwords were used in Ziegler et al., 2003). This allowed us to compare the amount of activation in potentially silent phonemes for the critical nonwords with respect to the control items. We also examined the overall mean number of silent-consonant responses that the model gave at different phoneme naming activation criterions. The results of the individual item activations appear in Fig. 6 and the mean results in Table 2.

As can be seen from the individual item results, compared to the control nonwords, there was less activation generated with the nonwords with potentially silent consonants, and there was an especially large amount of variability with the nonwords with extant bodies. Not surprisingly, given that CDP++ is affected by consistency (e.g., Perry et al., 2010), this suggests that whether CDP++ produces a final

Table 2
Proportion of final consonants produced as a function of the phoneme naming activation criterion.

Phoneme naming activation criterion	Proportion of final consonants produced			
	Extant body	Non-extant body	Extant body control	Non-extant body control
.47	.5	1	1	1
.52	.41	.91	1	1
.57	.35	.82	1	1
.62	.24	.82	1	1
.67	.18	.55	1	1
.72	.11	.36	1	1
.77	0	.18	.53	1

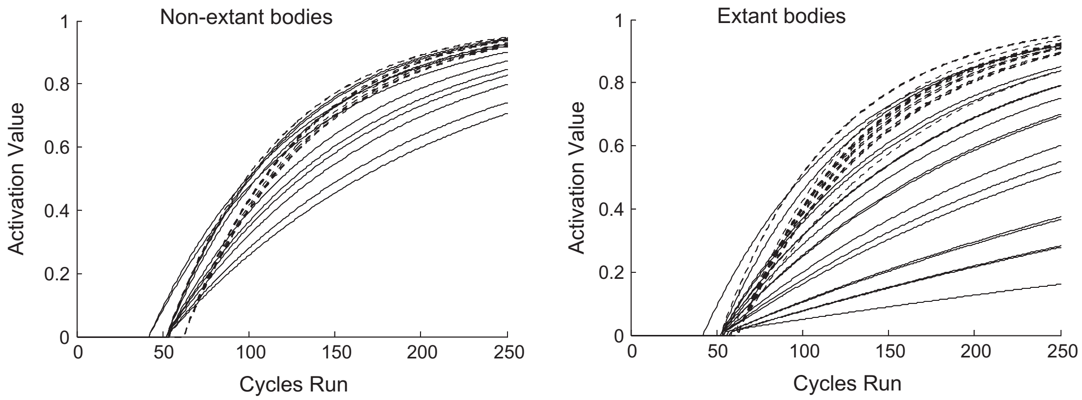

Fig. 6. Individual item activations of the final consonant in nonwords with non-extant and extant bodies and their controls. Note: Solid lines are the nonwords with non-extant and extant bodies and the dotted lines are the controls.

consonant is affected by other words that share the same body that the model is trained on. This can be seen from a correlation between the number of shared bodies taken from the training corpus (counting the nonwords with non-extant bodies as zero) and the final activation produced by the network, which had a value of $r = -.58$. Thus, the more silent-letter bodies a nonword shared with words, the more silent-consonants were given by the model. The same correlation run on the actual data instead of the CDP++ activations was $r = -.56$, suggesting that human readers are indeed affected by this factor. The item results also show why, in the mean data (see Table 2), whilst the number of silent final consonants increased as the phoneme naming activation criterion increased, this was largely restricted to the potentially silent consonant nonwords and not the control nonwords – the final consonant grapheme–phoneme relationships in the control nonwords were so well learned from words in the database that the activation values they produced were not only relatively homogenous but also relatively high. Thus, it was only when the phoneme naming activation criterion was set to a very high level that the model began to start using silent consonants with those nonwords.

The homogenous activations of the control nonwords compared to the critical nonwords and the greater activations found with the critical nonwords with non-extant vs. extant bodies shows that, as we predicted, final letters will tend to activate their phonemes, and this will be easier for the TLA network to learn than keeping them silent. In addition, as we also predicted, the –e on the end of control nonwords did not cause large amounts of activation across multiple phonemes. If this did occur, then the control nonwords should not have showed the pattern they did, as the spurious phonemes would have caused a lot of competition with the correct ones, and hence it should have been more difficult and slower for the final consonant to be activated. These results make a strong prediction, which is that even with stimuli sets that include very difficult items, people will still produce the final consonant in nonwords that end with a consonant-e pattern.

Theoretically, the results are interesting because the pattern somewhat resembles that of Andrews and Scarratt (1998) who found that, in English, even nonwords with relatively inconsistent bodies are generally given regular (i.e., rule-like) final pronunciations. Alternatively, they were able to find effects of orthographic bodies in specific instances where the nonwords used bodies that occurred in words whose pronunciations were never regular. With these nonwords, some but not all of the time, participants gave responses that could not be predicted using the most common grapheme–phoneme relationships. Although their data suggested that readers relied on a rule-based method with some exceptions where they used an analogy mechanism, this pattern was simulated later using CDP+ (Perry et al., 2007) which does not use a set of rules and an analogy mechanism, but rather a simple two-layer associative network. The results were also not driven by a mixing of lexical and sublexical activation, but rather learning in the sublexical route. The present results show a similar phenomenon: the network gave very rule-like responses to the control nonwords (i.e., "produce the consonant if there is an –e after it"), whereas it gave silent consonant responses to many nonwords that shared their bodies with words with silent consonants. Combined with the difference between these nonwords and the nonwords with non-extant bodies which had no pressure to be silent at the body-rime level, this suggests that spelling–sound consistency at both the grapheme–phoneme and body-rime level is important in French.

General discussion

We have developed a French version of CDP++.parser using the same assumptions as the English model (Perry et al., 2013) and examined whether the model could be extended to French. Such a development is important because the French orthography differs on a number of dimensions compared to other orthographies like English. These include it having a greater number of graphemes, many long sequences of vowel letters, and large numbers of commonly occurring letter sequences that do not appear to map to any sounds.

The model was evaluated in a number of domains. First, it was run on two large databases that exist in French to examine its quantitative performance. The results showed that on both databases examined, the model performed almost as well as a linear regression model that incorporated word frequency, orthographic wordlength, and orthographic neighborhood, although the actual amount of variance left once the articulatory characteristics of words were taken into account was not especially high. Nevertheless, this is an important benchmark (Balota & Spieler, 1998; Spieler & Balota, 1997), and one that only the CDP model family has yet been able to meet (Perry et al., 2007; Perry et al., 2010; Perry et al., 2013). Second, when evaluated on a number of small-scale benchmark effects using exactly the same parameter set as was used in the large-scale simulations, the model also performed well. It was able to capture the results found in a number of experiments examining spelling–sound consistency, as well as pseudohomophone effects.

One effect that the model failed to simulate was the orthographic neighborhood effect (Grainger, Spinelli, & Ferrand, 2000), an effect which has relatively low power (e.g., Spieler & Balota, 2000) and varies as a function of tasks, languages, and neighborhood manipulations (e.g., Andrews, 1997; Peeremen & Content, 1997; Ziegler & Perry, 1998). In the experiment we examined (Grainger et al., 2000), for example, neighborhood and RT were negatively related, but in the French database of Ferrand et al. (2011), there is a positive correlation with orthographic neighborhood if orthographic Levenshtein distances are added to the regression equation or a null effect if they are not. This effect is also a problem for the English CDP++. Given this, the best solution to this problem is likely to be a lexical route with somewhat different properties to the current one (e.g., Di Bono & Zorzi, 2013; Zorzi, Testolin, & Stoianov, 2013), and this may also help reduce the occasional implausible lexical capture the model makes. A second effect the model failed to capture was the syllable number effect with nonwords. Since the effect of orthographic neighborhood is likely to be at least in part lexical, the syllable number effect for nonwords is thus the only effect that the sublexical part of the model was not able to capture. However, as discussed above, it is actually possible that the syllable number effect arises at the level of articulatory planning or articulation (see Sulpizio et al., 2013), in which case it is beyond the scope of CDP++. Further research is needed to clarify this issue.

Finally, in terms of predicting patterns of generalization in nonword reading, which can be seen as a harder test than predicting response latencies (Perry et al., 2010), the model also showed reasonable results. Apart from having a relatively low overall error rate, the model also showed similar results to those found in an experiment examining the extent to which people generate phonology for consonants that lexically do not generally map to phonology. This pattern is peculiar to French, and the results of the experiment showed that people generally *do* produce phonology for these consonants, even when reading extant letter sequences where this never occurs in words. This pattern is interesting because it appears to show a deviation away from what might have been predicted via a simple statistical analysis of orthography–phonology relationships in a lexical corpus. These results provide further support to a basic tenet of the CDP model family, whereby the activation of sublexical phonology is the result of a direct (linear) mapping between spelling and sound that can be learnt via a simple associative learning mechanism.

Is the generation of sublexical phonology in CDP++ really linear?[5]

One of the fundamental differences between CDP++ and the connectionist models of the PDP family (e.g., Plaut et al., 1996) is the fact that CDP++ can only learn linear orthography-to-phonology mappings. It is precisely this *linear* learning mechanism which is assigned credit for explaining the present silent-letter results and also other aspects of reading (Perry et al., 2010) and reading development (Ziegler, Perry & Zorzi, 2014Ziegler et al., 2014). Whilst we have always claimed that CDP++ can only learn linear relationships, one could argue that non-linearity is introduced at the phoneme level by using a nonlinear squashing function. While this might appear to be a rather technical issue, it is one that is worth examining in detail given the importance of the learning mechanism in all of our simulation work.

In terms of how sublexical phonology is generated with the TLA (sublexical) network, the first computation is a simple feedfoward pass of the network. This works by presenting a simple vector to the model which represents the activation in the input nodes. A summed input for each node in the output of the network is then calculated by multiplying the strength of each weight by the input connected to the weight and then summing these (see also e.g., Baayen et al., 2011). This is clearly linear, since the value at each node is simply the sum of the weights connected to it multiplied by the input, and the inputs are entirely independent of each other, as are the outputs. This is identical to the way a simple linear regression works, with the summed input of a node being the predicted value and each weight being equivalent to a beta-value. Because of this, it means that any relationships that are ever *learned* by the network can only be linear. Unlike a simple regression equation, however, where the beta-values are determined via minimizing a sum of squares, the actual values of the weights are determined by an iterative learning procedure that is affected by the way error in the network is computed over time. In this case, error scores calculated in different ways, such as by using cross-entropy rather than simple error differences (e.g., Plaut et al., 1996) or by adding error to simulate atypical learning trajectories (i.e., dyslexia; see Ziegler et al., 2014), affect the final weights.

The second computation of the TLA network is how a response is generated once the summed input for each output phoneme is calculated. The model first puts the sums at each output node into exactly the same function as described in Zorzi et al. (1998, Eq. (2): $Oi = \frac{1}{1+e^{-(neti-1)\tau}}$ where net_i is the summed input into the function, and τ is the temperature which was set to 3 like all other versions of CDP). This function takes the summed activation as the parameter of an S-shaped squashing function that bounds the input between zero and one, with no input causing no output (i.e., zero output). Because this function is monotonically increasing, the actual rank order of activations produced by the network is identical to the rank order of the activations produced in the first step of the computation where only summed values were computed. Thus, if a pronunciation was produced after this calculation had been done by simply choosing a level at which a phoneme was considered activated, identical results could be found (albeit with a different phoneme naming activation criterion, which specifies the level at which a phoneme must be actived to get into the final pronunciation). The main effect that this change has is therefore on learning. In this case, when the activation is produced by the network and compared to the target activation, the error term is not computed with a simple linear function, but is instead computed from a sigmoid function that is of a similar shape to that used in logistic regression (e.g., Tabachnik & Fidell, 2001). This means that the final outputs of the network after being transformed by the sigmoid function will tend to approximate what a set of logistic regressions would produce rather than a set of simple regressions. Thus, the network learns linear relationships between the predictors (graphemes) and the sigmoid transformation of the outputs (phonemes). This has very little effect on what is learned by the network compared to a purely linear function. In particular, as can be seen from the Supplementary materials, where we trained and ran a network without a sigmoid function, the results are almost the same as those the standard network that uses a sigmoid function produces, including the pronunciations of nonwords at the individual item level.

Adequacy of representations

One final issue that needs to be considered is the organization of letters used in the model. In all versions of CDP,

[5] We thank Dave Plaut for suggesting we consider this.

simple contiguous letter strings have been used. However, there is some recent evidence (Chetail & Content, 2012; Chetail & Content, 2013) that the letter level representations used in French may not be simple contiguous strings and, in some cases, may not be isomorphic with phonology either. The evidence that Chetail and Content used to infer this came from hiatus and schwa words. Hiatus words differ from most other words in that they have a string of contiguous vowels in them where the vowels map to phonemes in two syllables. For example, with the word *chaos*, there is an –ao vowel sequence where the –a appears to map to the vowel of the first syllable and the –o to the second. Across a number of experiments, Chetail and Content (2012) found that hiatus words were more difficult to process than control words. They argued that this was caused by the initial organization of letters, where letters are grouped into clusters of consonants and vowels, regardless of the phonological form (see the Supplementary materials for a simulation of the only naming experiment they did). Thus *chaos* would be a single orthographic syllable organized into consonants and vowels even though it has two phonological syllables (/ka.o/), and this causes difficulties in processing. In terms of the schwa words, these were so named because they contain an orthographic sequence with an –e that often maps to a schwa (e.g., the first –e in *simplement* /sɛ̃.plə̲.mɑ̃). Chetail and Content (2013) examined these words except they used ones that, whilst being of a similar structure to other schwa words, used an –e that did not obviously map to any phonological vowel. For example, the –e in *biberon* (/bi.bʀô/) looks like it is embedded within letters that appear to represent an onset cluster (i.e., *-ber*, which maps to the phonology /bʀ/). The results Chetail and Content found from a simple segmentation task where participants marked where they thought syllable segmentations should go and a letter search task suggested that that the –e may in fact form the end of an orthographic syllable, with the letter after it occurring as an onset. Thus, the word *biberon* would use the orthographic segmentation bi.be.ron, which has three orthographic syllables, even though its phonology only has two syllables.

Whilst the data from these experiments may appear difficult for CDP++ to reconcile with the sublexical route, which is trained on letters segmented into phonological syllables, it is in fact within the scope of the model to explain. This is because the objective of the sublexical route is to learn the relationships between graphemes and output phonology, which means that data to do with different types of *letter* groupings is not in conflict with the model, and indeed other groupings have been proposed, such as purely consonants and vowels (e.g., Perea & Lupker, 2004), open-bigrams (Grainger & Van Heuven, 2003), and non-phonological syllabic structures (e.g., Taft & Krebs-Lazendic, 2013). In particular, with CDP++, pure letter information is generated at the letter level, and the model is largely agnostic as to exactly how this occurs, with the current representations being used largely for convenience (although consonant/vowel information about letters could be used by the grapheme parser; Perry et al., 2010). The main assumption in CDP++ is that letters at this level are processed quickly and automatically, and that activation at this level can flow to the orthographic lexicon or trigger the start of the graphemic parsing process. The graphemic parsing process, first introduced in Perry et al. (2007), selects individual letters from potentially competing alternatives once letter information becomes available. This is not an automatic process, but rather requires focused visuo-spatial attention whilst the parser selects graphemes from the letters and places them in the graphemic buffer. Assuming letter processing is very automatic, minor differences in the way different letters are activated is largely irrelevant to the parser. One reason for this is because the selection process simply chooses each letter that has the highest activation amongst potential alternatives, and thus absolute activation differences between letters makes essentially no difference. A second reason is that the graphemic parsing process is relatively slow compared to letter processing, which means that letters would almost always be available for the model to select once the initial letter has triggered the start of the parsing process.

Conclusions

We have implemented and tested a full-blown model of multisyllabic reading aloud for French. The model learns to select graphemes and learns grapheme-to-phoneme mappings. It was tested on extant data and an additional experiment examining the reading aloud of nonwords with potentially silent letters. The results of the silent-letter experiment showed that, contrary to what would be predicted on the basis of lexical database statistics, people generally pronounce "silent" consonants in nonwords. We showed that the French CDP++ model faithfully predicted this effect because it implements a *linear* mapping between orthography and phonology. These findings highlight the theoretical and practical significance of using computational models to help determine the processes and representations that underlie skilled reading.

Acknowledgments

This research was supported by grants from the Australian Research Council (DP120100883) to CP and the European Research Council (210922-GENMOD) to MZ. Part of this work was performed on the swinSTAR supercomputer at Swinburne University of Technology. The French CDP++ is available for download at http://ccnl.psy.unipd.it/CDP.html and https://sites.google.com/site/conradperryshome/. We would like to thank Max Coltheart, Alain Content, Fabienne Chetail, and David Plaut for suggestions which made this paper much better.

Appendix A

The database

The database we used was Lexique 3.62 (New et al., 2001). This database has 128,918 words in it. From these, only words of 3 syllables or less were used, and words with characters that were not in the standard French alphabet were removed. This left 101,396 words that were made up of 37 letters and 38 phonemes.

Deriving the graphemes

To derive the graphemes that are used in words, apart from as mentioned above about coda graphemes, a very similar procedure as Perry et al. (2013) was used, where a set of graphemes were aligned with phonemes in a one-grapheme–one-phoneme fashion where possible. This was done in the context of the 111 graphemes that were chosen to code the database. These appear in the Supplementary materials. The following strategy was used to code the database so that graphemes in the orthographic template were placed in the same place as phonemes in the phoneme output buffer where possible:

(a) When the phonology of two consonants occurred together in an onset cluster but where the written form was separated orthographically by an –e (*adopterais* - /adɔptre/), the –e was placed in an onset position (i.e., the -t.e.r graphemes were used as the onset of the third syllable).
(b) The vowels, a, e, i, ï, o, u, ou can function as either semi-vowels in onset positions or as vowels. These were coded as being in an onset or vowel category based on this distinction.
(c) -y was coded as an onset when it mapped to a /j/ in an onset position (e.g., *employait* – /ɑ~.plwa.je/) or a vowel if it did not (e.g., *cycle* – /cikl/)

After the alignment procedure, there were 329 words where coding appeared difficult in terms of the strategy of aligning graphemes with phonemes and the algorithm used to do it. Many of these were loan words (e.g., *beagle*), acronyms (e.g., *rna*), letter names (e.g., *w*), and Roman numerals (e.g., *lxiv*). This left 101067 words for training.

Training the graphemic parser

From the training words, 707,676 individual training exemplars representing the way parts of words were presented to the graphemic parser were constructed. These specified the letter sequences that each grapheme in each word occurs in as they should be chosen by a parser with an attentional window of 5 letters (e.g., a word like *chef* would create 3 patterns that the parser would learn from since it has 3 graphemes and hence the parser would need to learn to select and classify the 3 graphemes (i.e., chef* (ch, onset), ef*** (e, vowel), f**** (f, coda)).

After the database was constructed, a network that was identical to that in Perry et al. (2013) excluding the number of graphemes and phonemes was trained for 15 cycles. The parameters used were also identical to Perry et al. The learning rate (.05) of each exemplar was multiplied by the value created by taking its log word frequency plus 2 and dividing it by the log word frequency of the highest frequency word in the database (which was 867,041) plus 2. The plus 2 was used because some words have zero frequency in the database. This was done with all of the words in the database as well as with a number of smaller subsets (500, 1000, 2000, and 5000 words) that were sampled in frequency order, where higher frequency words were chosen over lower frequency words. Once the models were fully trained, they were tested on all of the patterns. The results of the model on all individual exemplars (i.e., whether the model chose each possible grapheme in each possible word correctly) and with words, where words were counted correct only if all graphemes in them were correctly parsed, appears in Fig. A1.

As can be seen, the model displayed an overall error rate that was low enough for modelling purposes (.51% on graphemes; 3.27% on words) – indeed lower than the English model. The model was also able to perform reasonably well even when trained on a very limited number of exemplars. The model trained on 5000 words (4.93% of the database), for example, had an error rate after training of 3.34% and 21.21% when tested on all exemplars and words in the database.

Apart from the overall error rate, it is also important to see whether the model makes any systematic errors unlike those that people do (e.g., choosing graphemes that would cause phonotactically illegal pronunciations). One likely source of these are graphemes that are used in different

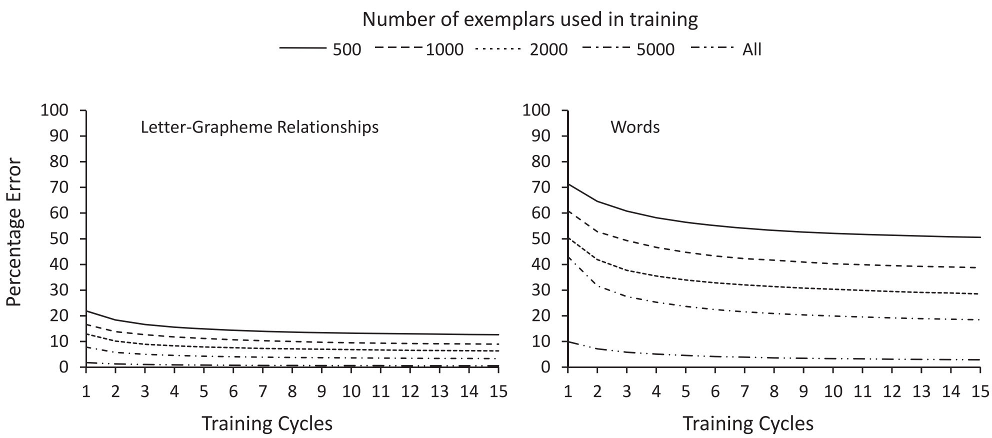

Fig. A1. Performance of the graphemic parser on individual letter–grapheme relationships and words using different numbers of training exemplars.

categories (i.e., onset, vowel, and coda), because selecting the correct graphemes needs to be done based on the contextual information of other letters around the grapheme. To investigate this, we examined the following graphemes: a, e, i, ï, o, ou, u, y. The results showed that the error rate with the graphemes was still low (.12%, 1.02%, .91%, 4.44%, .41%, .48%, .29%, .91%, respectively). Thus, even though the way these graphemes are used in French is relatively complex, it did not appear to stop the network learning the relationships needed to categorize them correctly.

Apart from the overall results, since the –e may be used in onset, vowel, and coda positions, it represents an especially difficult challenge for the model, and so the performance of the model was examined in these three different categories separately. The results showed that the model had an error rate of .82%, .18%, and .02% on the onset (e.g., *cerise*), vowel (e.g., *obtenir*), and coda graphemes (e.g., *couennes*). This means that the network was reasonably accurate at categorizing the –e correctly in all major contextual situations that it occurs in (i.e., in onset (e.g., *adopterais*), vowel (e.g., *tres*), and coda positions (e.g., *visage*)). This suggests the model can learn how graphemes should be categorized based on the letter context they appear in, and that it does not simply default to the highest frequency alternative (-e is used as an onset only 6.43% of the time). A similar analysis performed on the –i grapheme, which commonly maps to phonemes in both vowel and onset positions, also showed that the model did not simply choose the most common categorization, with the error rate being .68% and .23% for the vowel and onset categories, respectively.

CDP++. details

The lexical route

In terms of the lexical route, to simplify matters, since there can be 27 slots in the output (i.e., the phoneme output buffer), we also used 27 slots in the input for both the features and the letters. It would have been possible to use only 16 slots, since the longest word in the database is 16 letters long and an essentially identical performance could have been obtained if only 16 letters were used (note that it is possible to create fairly strange looking nonwords that have more than 16 letters). We make no strong claims about the theoretical relevance of this, and it is assumed that, like all of the CDP models, given how crude the front end of the model is, this simply represents a convenient starting point for the model rather than something that we intend to make serious theoretical claims about. Given this, the features for the feature level of the model were chosen arbitrarily, although the parameters were set so that there was essentially no effect of overlap caused by differences between features in letters and the way they activate the letter level.

In terms of the other parts of the model, the same frequency counts were used in both the orthographic and phonological lexicons since the database we used only had one set of counts. In addition, all of the words use a frequency count of those given in the database plus 2. This was done because some words have a frequency value of zero and we take log values of frequencies for some computations. This means all frequency values always end up being greater than 0.

The TLA network

The TLA network was trained on words organized into their grapho-syllabic structure based on the graphemes that the words were initially decomposed into. Thus there were 101,067 training patterns that specified the graphemes in words organized into a syllabic structure, as well as their phonology. The order of these training patterns was first randomized, and then the model was trained for 20 cycles. Identical training parameters as the graphemic parser were used. Like the graphemic parser, the learning rate of each word was multiplied by the value created by taking its log word frequency plus 2 and dividing it by the log word frequency plus 2 of the highest frequency word in the database.

The parameters used in running mode appear in the Supplementary materials.

Appendix B

Individual item statistics for the nonwords used, CDP++, and DRC.

Nonword	Body type	N. shared bodies	Results (final phonemes)			Results (RTs)		
			Actual % non-silent	CDP++	DRC	Actual	CDP++	DRC
nart	Extant	5	65.52	Prod	Silent	593	186	128
mard	Extant	10	50.00	Silent	Silent	589	166	111
fers	Extant	7	36.67	Silent	Silent	633	112	103
bez	Extant	4	89.66	Silent	Silent	605	121	115
lant	Extant	7	56.25	Silent	Silent	541	113	115
lond	Extant	7	21.88	Prod	Silent	553	198	113
naid	Extant	1	79.31	Prod	Prod	662	171	118
nert	Extant	2	57.14	Prod	Silent	589	170	110

(*continued on next page*)

Appendix B (*continued*)

Nonword	Body type	N. shared bodies	Results (final phonemes)			Results (RTs)		
			Actual % non-silent	CDP++	DRC	Actual	CDP++	DRC
mied	Extant	2	59.26	Prod	Silent	629	148	137
lont	Extant	17	41.94	Silent	Silent	556	111	113
puid	Extant	1	50.00	Prod	Silent	600	124	107
noid	Extant	1	43.33	Silent	Silent	604	197	113
lurs	Extant	6	59.38	Silent	Silent	617	118	127
lirs	Extant	2	53.33	Silent	Silent	643	115	107
nuid	Extant	1	50.00	Silent	Silent	631	196	108
biz	Extant	1	100.00	Prod	Silent	571	118	109
lint	Extant	2	59.00	Silent	Silent	591	137	117
taup	Non-Extant	0	89.29	Prod	Prod	600	114	107
nirt	Non-Extant	0	90.00	Prod	Silent	609	168	128
nurt	Non-Extant	0	55.56	Prod	Silent	662	150	128
murd	Non-Extant	0	77.42	Prod	Silent	621	190	107
mird	Non-Extant	0	75.00	Silent	Silent	633	198	113
buz	Non-Extant	0	96.55	Prod	Prod	590	117	102
boz	Non-Extant	0	96.77	Prod	Prod	606	125	111
lunt	Non-Extant	0	83.33	Prod	Prod	631	141	118
pind	Non-Extant	0	66.67	Prod	Prod	659	181	118
pund	Non-Extant	0	85.19	Prod	Prod	745	138	118
neid	Non-Extant	0	78.57	Prod	Prod	622	157	118
			% Non-Silent					
Average (extant body)			57.21	41	6	601	147	115
Average (non-extant body)			81.30	91	64	634	153	115

Note: Prod = Consonant produced; RTs = Reaction Times.

C. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jml.2014.01.003.

References

Andrews, S. (1997). The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin & Review, 4*, 439–461. http://dx.doi.org/10.3758/BF03214334.

Andrews, S., & Heathcote, A. (2001). Distinguishing common and task-specific processes in word identification: A matter of some moment? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 27*, 514–544. http://dx.doi.org/10.1037/0278-7393.27.2.514.

Andrews, S., & Scarratt, D. R. (1998). Rule and analogy mechanisms in reading nonwords: Hough Dou Peapel Rede Gnew Wirds? *Journal of Experimental Psychology: Human Perception and Performance, 53*, 567–593. http://dx.doi.org/10.1037/0096-1523.24.4.1052.

Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review, 105*, 678–723. http://dx.doi.org/10.1037/0033-295X.105.4.678-723.

Baayen, R. H., Milin, P., Durdevic, D. F., Hendrix, P., & Marelli, M. (2011). An amorphous model for morphological processing in visual comprehension based on naive discriminative learning. *Psychological Review, 118*, 438–481. http://dx.doi.org/10.1037/a0023851.

Balota, D. A., & Spieler, D. (1998). The utility of item-level analysis in model evaluation: A reply to Seidenberg and Plaut. *Psychological Science, 9*, 238–240. http://dx.doi.org/10.1111/1467-9280.00047.

Caramazza, A., & Miceli, G. (1990). The structure of graphemic representations. *Cognition, 37*, 243–297. http://dx.doi.org/10.1016/0010-0277(90)90047-N.

Chetail, F., & Content, A. (2012). The internal structure of chaos: Letter category determines visual word perceptual units. *Journal of Memory and Language, 67*, 371–388. http://dx.doi.org/10.1016/j.jm1.2012.07.004.

Chetail, F., & Content, A. (2013). Segmentation of written words in French. *Language and Speech, 56*, 125–144. http://dx.doi.org/10.1177/0023830912442919.

Cohen, J. D., MacWhinney, B., Flatt, M., & Provost, J. (1993).). PsyScope: An interactive system for graphic designing and controlling experiments in the psychology laboratory using Macintosh computers. *Behavioral Research Methods, Instruments, & Computers, 25*, 252–271. http://dx.doi.org/10.3758/BF03204507.

Coltheart, M., Davelaar, E., Jonasson, J., & Besner, D. (1977). Access to the internal lexicon. In S. Domic (Ed.), *Attention and performance VI* (pp. 535–555). Hillsdale, NJ: Erlbaum.

Coltheart, M., Rastle, K., Perry, C., Langdon, R., & Ziegler, J. C. (2001). DRC: A computational model of visual word recognition and reading aloud. *Psychological Review, 108*, 204–256. http://dx.doi.org/10.1037/0033-295X.108.1.204.

Content, A. (1991). The effect of spelling-to-sound regularity on naming in French. *Psychological Research, 53*, 3–12. http://dx.doi.org/10.1007/BF00867327.

Cotelli, M., Abutalebi, J., Zorzi, M., & Cappa, S. F. (2003). Vowels in the buffer: A case study of acquired dysgraphia with selective vowel substitutions. *Cognitive Neuropsychology, 20*, 99–114. http://dx.doi.org/10.1080/02643290244000158.

Courrieu, P., Brand-D'abrescia, M., Peereman, R., Spieler, D., & Rey, A. (2011). Validated intraclass correlation statistics to test item performance models. *Behavioral Research Methods, 43*, 37–55. http://dx.doi.org/10.3758/s13428-010-0020-5.

Di Bono, M. G., & Zorzi, M. (2013). Deep generative learning of location-invariant visual word recognition. *Frontiers in Psychology, 4*.

Dupoux, E., Pallier, C., Sebastián, N., & Mehler, J. (1997). A destressing "deafness" in French? *Journal of Memory and Language, 36*, 406–421.

Ferrand, L. (2000). Reading aloud polysyllabic words and nonwords: The syllabic length effect reexamined. *Psychonomic Bulletin & Review, 7*, 142–148.

Ferrand, L., Brysbaert, M., Keuleers, E., New, B., Bonin, P., Méot, A., et al. (2011). Comparing word processing times in naming, lexical decision, and progressive demasking: Evidence from Chronolex. *Frontiers of Psychology, 2*, 306. http://dx.doi.org/10.3389/fpsyg.2011.00306.

Frost, R. (2012). A universal approach to modeling visual word recognition and reading: Not only possible, but also inevitable.

Behavioral and Brain Sciences, 35, 310–329. http://dx.doi.org/10.1017/S0140525X12000635.

Grainger, J., Spinelli, D., & Ferrand, L. (2000). Effect of baseword frequency and orthographic neighborhood size in psudohomophone naming. *Journal of Memory and Language, 42*, 88–102. http://dx.doi.org/10.1006/jmla.1999.2675.

Grainger, J., & Van Heuven, W. (2003). Modeling letter position coding in printed word perception. In P. Bonin (Ed.), *The mental lexicon* (pp. 1–24). New York: Nova Science Publishers.

Howard, D., & Best, W. (1996). Developmental phonological dyslexia: Real word reading can be completely normal. *Cognitive Neuropsychology, 13*, 887–934. http://dx.doi.org/10.1080/026432996381854.

Jared, D. (2002). Spelling–sound consistency and regularity effects in word naming. *Journal of Memory and Language, 46*, 723–750. http://dx.doi.org/10.1006/jmla.2001.2827.

Kinoshita, S., & Lupker, S. J. (2003). Priming and attentional control of lexical and sublexical pathways in naming: A reevaluation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 29*, 405–415. http://dx.doi.org/10.1037/0278-7393.29.3.405.

Meyer, A. S., Roelofs, A., & Levelt, W. J. M. (2003). Word length effects in object naming: The role of a response criterion. *Journal of Memory and Language, 48*, 131–147. http://dx.doi.org/10.1016/S0749-596X(02)00509-0.

New, B., Pallier, C., Ferrand, L., & Matos, R. (2001). Une base de données lexicales du français contemporain sur Internet: LEXIQUE. *L'Année Psychologique, 101*, 447–462. http://dx.doi.org/10.3406/psy.2001.1341.

Norris, D. (2013). Models of word recognition. *Trends in Cognitive Sciences, 13*, 517–524. http://dx.doi.org/10.1016/j.tics.2013.08.003.

Peeremen, R., & Content, A. (1997). Orthographic and phonological neighborhoods in naming: Not all neighbors are equally influential in orthographic space. *Journal of Memory and Language, 37*, 382–410. http://dx.doi.org/10.1006/jmla.1997.2516.

Perea, M., & Lupker, S. J. (2004). Can CANISO activate CASINO? Transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory and Language, 51*, 231–246. http://dx.doi.org/10.1016/j.jml.2004.05.005.

Perry, C. (2013). Graphemic parsing and the basic orthographic syllable structure. *Language and Cognitive Processes, 28*, 355–376. http://dx.doi.org/10.1080/01690965.2011.641386.

Perry, C., Ziegler, J. C., Braun, M., & Zorzi, M. (2010). Rules versus statistics in reading aloud: New evidence on an old debate. *European Journal of Cognitive Psychology, 22*, 798–812. http://dx.doi.org/10.1080/09541440902978365.

Perry, C., Ziegler, J. C., & Zorzi, M. (2007). Nested modeling and strong inference testing in the development of computational theories: The CDP+ model of reading aloud. *Psychological Review, 27*, 301–333. http://dx.doi.org/10.1037/0033-295X.114.2.273.

Perry, C., Ziegler, J. C., & Zorzi, M. (2010). Beyond single syllables: Large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model. *Cognitive Psychology, 61*, 106–151. http://dx.doi.org/10.1016/j.cogpsych.2010.04.001.

Perry, C., Ziegler, J. C., & Zorzi, M. (2013). A computational and empirical investigation of graphemes in reading. *Cognitive Science, 37*, 800–828. http://dx.doi.org/10.1111/cogs.12030.

Plaut, D. C., McClelland, J. L., Seidenberg, M. S., & Patterson, K. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review, 103*, 56–115. http://dx.doi.org/10.1037/0033-295X.103.1.56.

Prinzmetal, W., Treiman, R., & Rho, S. H. (1986). How to see a reading unit. *Journal of Memory and Language, 25*, 461–475. http://dx.doi.org/10.1016/0749-596X(86)90038-0.

Pritchard, S. (2013). *Incorporating learning mechanisms into the dual-route cascaded (DRC) model of reading aloud and word recognition*. Doctor of Philosophy, Macquarie University.

Rey, A., Ziegler, J. C., & Jacobs, A. M. (2000). Graphemes are perceptual reading units. *Cognition, 75*, B1–B12. http://dx.doi.org/10.1016/S0010-0277(99)00078-5.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review, 96*, 523–568. http://dx.doi.org/10.1037/0033-295X.96.4.523.

Share, D. L. (2008). On the Anglocentricities of current reading research and practice: The perils of overreliance on an "outlier" orthography. *Psychological Bulletin, 134*, 584–615. http://dx.doi.org/10.1037/0033-2909.134.4.584.

Spieler, D. H., & Balota, D. A. (1997). Bringing computational models of word naming down to the item level. *Psychological Science, 8*, 411–416. http://dx.doi.org/10.1111/j.1467-9280.1997.tb00453.x.

Spieler, D. H., & Balota, D. A. (2000). Factors influence word naming in younger and older adults. *Psychology and Aging, 15*, 225–231. 10.1037∗0882-7974.15.2325.

Sulpizio, S., Arduino, L. S., Paizi, D., & Burani, C. (2013). Stress assignment in reading Italian polysyllabic pseudowords. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 39*, 51–68. http://dx.doi.org/10.1037/a0028472.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88*, 135–170. http://dx.doi.org/10.1037/0033-295X.88.2.135.

Tabachnik, B. G., & Fidell, L. S. (2001). *Using multivariate statistics* (4th ed.). Needhham Heights, MA: Allyn & Bacon.

Taft, M. (2001). Processing of orthographic structure by adults of different reading ability. *Language and Speech, 44*, 351–376. http://dx.doi.org/10.1177/00238309010440030301.

Taft, M., & Krebs-Lazendic, L. (2013). The role of orthographic syllable structure in assigning letters to their position in visual word recognition. *Journal of Memory and Language, 68*, 85–97. http://dx.doi.org/10.1016/j.jml.2012.10.004.

Tainturier, M. J., & Rapp, B. C. (2004). Complex graphemes as functional spelling units: Evidence from acquired dysgraphia. *Neurocase, 10*, 122–131. http://dx.doi.org/10.1080/13554790490497238.

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

Widrow, G., & Hoff, M. E. (1960). *Adaptive switching circuits. Institute of radio engineers, western electronic show and convention record, Part 4.* New York: Institute of Radio Engineers (pp. 96–104). New York: Institute of Radio Engineers.

Yang, J. F., McCandliss, B. D., Shu, H., & Zevin, J. (2009). Simulating language-specific and language-general effects in a statistical learning model of Chinese reading. *Journal of Memory and Language, 61*, 238–257. http://dx.doi.org/10.1016/j.jml.2009.05.001.

Yarkoni, T., Balota, D., & Yap, M. (2008). Moving beyond Coltheart's N: A new measure of orthographic similarity. *Psychonomic Bulletin & Review, 15*, 971–979. http://dx.doi.org/10.3758/PBR.15.5.971.

Ziegler, J. C., & Goswami, U. (2006). Becoming literate in different languages: Similar problems, different solutions. *Developmental Science, 9*, 429–453. http://dx.doi.org/10.1111/j.1467-7687.2006.00509.x.

Ziegler, J. C., & Perry, C. (1998). No more problems in Coltheart's neighborhood: Resolving neighborhood conflicts in the lexical decision task. *Cognition, 68*, B53–B62. http://dx.doi.org/10.1016/S0010-0277(98)00047-X.

Ziegler, J. C., Perry, C., & Coltheart, M. (2003). Speed of lexical and nonlexical processing in French: The case of the regularity effect. *Psychonomic Bulletin & Review, 10*, 947–953. http://dx.doi.org/10.3758/BF03196556.

Ziegler, J. C., Perry, C., & Zorzi, M. (2014). Modelling reading development through phonological decoding and self-teaching: Implications for dyslexia. *Philosophical Transactions of the Royal Society B: Biological Sciences, 369*. http://dx.doi.org/10.1098/rstb.2012.0397. Art. No. 20120397.

Zorzi, M. (2010). The connectionist dual process (CDP) approach to modeling reading aloud. *The European Journal of Cognitive Psychology, 22*, 836–860. http://dx.doi.org/10.1080/09541440903435621.

Zorzi, M., Houghton, G., & Butterworth, B. (1998). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance, 24*, 1131–1161. http://dx.doi.org/10.1037//0096-1523.24.4.1131.

Zorzi, M., Testolin, A., & Stoianov, I. P. (2013). Modeling language and cognition with deep unsupervised learning: A tutorial overview. *Frontiers in Psychology, 4*. http://dx.doi.org/10.3389/fpsyg.2013.00515 (Article 515).