



ELSEVIER

Contents lists available at ScienceDirect

## Cognitive Psychology

journal homepage: [www.elsevier.com/locate/cogpsych](http://www.elsevier.com/locate/cogpsych)



# Beyond single syllables: Large-scale modeling of reading aloud with the Connectionist Dual Process (CDP++) model

Conrad Perry<sup>a,\*</sup>, Johannes C. Ziegler<sup>b</sup>, Marco Zorzi<sup>c</sup>

<sup>a</sup> Faculty of Life and Social Sciences, Swinburne University of Technology, Australia

<sup>b</sup> Laboratoire de Psychologie Cognitive, Aix-Marseille Université and Centre National de la Recherche Scientifique, Marseille, France

<sup>c</sup> Dipartimento di Psicologia Generale and Center for Cognitive Science, Università di Padova, Italy

## ARTICLE INFO

### Article history:

Accepted 13 April 2010

Available online 26 May 2010

### Keywords:

Reading aloud  
Computational modeling  
Disyllables  
Word stress

## ABSTRACT

Most words in English have more than one syllable, yet the most influential computational models of reading aloud are restricted to processing monosyllabic words. Here, we present CDP++, a new version of the Connectionist Dual Process model (Perry, Ziegler, & Zorzi, 2007). CDP++ is able to simulate the reading aloud of mono- and disyllabic words and nonwords, and learns to assign stress in exactly the same way as it learns to associate graphemes with phonemes. CDP++ is able to simulate the monosyllabic benchmark effects its predecessor could, and therefore shows full backwards compatibility. CDP++ also accounts for a number of novel effects specific to disyllabic words, including the effects of stress regularity and syllable number. In terms of database performance, CDP++ accounts for over 49% of the reaction time variance on items selected from the *English Lexicon Project*, a very large database of several thousand of words. With its lexicon of over 32,000 words, CDP++ is therefore a notable example of the successful scaling-up of a connectionist model to a size that more realistically approximates the human lexical system.

© 2010 Elsevier Inc. All rights reserved.

## 1. Introduction

Most words in English have more than one syllable (e.g., Baayen, Piepenbrock, & van Rijn, 1993), yet the most influential computational models of reading aloud have been developed for monosyllabic

\* 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](mailto:ConradPerry@gmail.com) (C. Perry).

words (e.g., Coltheart, Rastle, Perry, Langdon, & Ziegler, 2001; Perry, Ziegler, & Zorzi, 2007; Plaut, McClelland, Seidenberg, & Patterson, 1996; Seidenberg & McClelland, 1989; Zorzi, Houghton, & Butterworth, 1998a). Jared and Seidenberg (1990) noted this imbalance almost two decades ago, stating: “Although a great deal is known about the naming process, a serious limitation of previous work is that it has been largely concerned with the processing of monosyllabic words” (p. 92). This situation has not changed much since.

It is possible to identify at least three reasons why the modeling of multisyllabic word reading has lagged behind. First, many modelers would argue that a sensible modeling strategy is to “start small”, thus reducing the complexity of the models and allowing an in-depth understanding of their fundamental properties (see e.g., Becker, Behrmann, Moscovitch, & Joordens, 1997; Kawamoto & Zemblidge, 1992; Perry, 1999). Second, the empirical database of English has been strongly biased towards monosyllabic words. That is, many landmark investigations of the classic benchmark effects were initially done using monosyllabic words (e.g., Glushko, 1979; Jared, 2002; Taraban & McClelland, 1987; Weekes, 1997). Similarly, the highly influential work based on regression analyses of large-scale databases was initially focused on monosyllables (Balota, Cortese, Sergent-Marshall, Spieler, & Yap, 2004; Spieler & Balota, 1997; Treiman, Mullenix, Bijeljac-Babic, & Richmond-Welty, 1995). Finally, and most importantly, modeling the reading aloud of multisyllabic words is simply a more difficult enterprise because a number of additional issues have to be addressed, such as syllabification strategies and stress assignment.

To give an example of the kind of problems that have to be dealt with when reading multisyllabic words, consider the words *cancer* and *canal*. With these words, any model of disyllabic reading not only needs to know where to put the syllable boundary (*can.cer* versus *ca.nal*), but also that *cancer* is stressed on the first syllable and *canal* is stressed on the second. Even assuming that it is possible to look up this information in a phonological lexicon, one still faces the problem that people can read nonwords, such as *commoke* or *zortess* (see Rastle & Coltheart, 2000), for which stress is consistently assigned on the first syllable in *zortess* and on the second in *commoke*. This means that in the absence of lexical phonology, people are able to assign stress nonlexically. Any new model of disyllabic reading aloud should be able to predict such patterns.

In the present paper, we present a new computational model of disyllabic word reading. In the spirit of the nested incremental modeling strategy advocated in our previous work, this model is an extension of the Connectionist Dual Process (CDP) model (Perry et al., 2007; Perry, Ziegler, Braun, & Zorzi, 2010; Zorzi et al., 1998a; see Zorzi (2010) for a review). At present, the most recent version (CDP+) has been shown to be the most successful model of reading aloud, at least in terms of its quantitative performance on monosyllabic words. We refer to the new model as CDP++ because it includes its own precursor (CDP+) as a special case. We start with an overview of the benchmark effects that any computational model of reading aloud that deals with disyllabic words should be able to address and then briefly discuss the two existing computational models that can simulate the reading aloud of multisyllabic words. Finally, we give a full description of CDP++ and present a thorough evaluation of its performance against a number of benchmark effects described below.

### 1.1. Benchmarks for a model of disyllabic word reading

#### 1.1.1. Monosyllabic word reading

Any model of disyllabic word reading should be able to simulate the critical empirical phenomena identified for monosyllabic word reading – that is, it should be *backwards compatible*. Backwards compatibility is a key element in incremental nested modeling (Jacobs & Grainger, 1994). A list of monosyllabic benchmark effects has been proposed by Perry et al. (2007, Table 4). This list includes the effects of word frequency (e.g., Weekes, 1997), spelling-sound consistency (e.g., Jared, 2002), and word length (e.g., Ziegler, Perry, Jacobs, & Braun, 2001), as well as various interactions between these factors. Moreover, the model needs to be able to read nonwords with a high level of accuracy (e.g., Besner, Twilley, McCann, & Seergobin, 1990) and give pronunciations that are similar to those given by skilled readers (e.g., Andrews & Scarratt, 1998; Seidenberg, Plaut, Petersen, McClelland, & McCrae, 1994).

### 1.1.2. Large-scale database performance

In contrast to the thousands of experiments on monosyllabic word reading, the number of experiments examining the reading aloud of disyllabic words is very limited (see below). Fortunately, however, the *English Lexicon Project* (ELP; Balota et al., 2007) includes data from 1200 skilled adult readers on more than 40,000 words, many of which are disyllabic, and thus provides a rich and very large database of reading performance. This database has been used to evaluate model performance at the item-level by regressing model latencies onto the human ones (Yap & Balota, 2009). This provides an overall goodness-of-fit measure in terms of the percentage of variance accounted for, which is a sensitive measure for adjudicating between existing models (Coltheart et al., 2001; Perry et al., 2007).

A second way the ELP database has been used is by examining theoretically important factors that models are sensitive to (e.g., frequency, length, consistency, etc.), rather than just overall fit. This was done by Yap and Balota (2009) who used a hierarchical regression approach in which different variables of theoretical interest were entered as predictors of both human and model naming latencies. This approach allowed them to investigate the extent to which a model is sensitive to the same variables that influence human performance both in terms of the strength and the direction of the effects. In particular, Yap and Balota first examined a group of variables that was designed to account for variance in the onsets of words, they then added the effect of word stress, followed by a set of standard lexical variables, and finally a number of more intricate variables such as those to do with sublexical orthography–phonology mappings.

Apart from the effects of single variables, there are also a number of interactions that show systematic effects in the reading aloud of disyllabic words. In particular, Yap and Balota (2009) examined how frequency interacts with a number of theoretically interesting variables, including syllable number, letter length, orthographic neighborhood, and spelling–sound consistency. In small-scale experiments, the effect of these variables has often been found to greatly diminish when examined with high-frequency words (e.g., Andrews, 1989; Ferrand, 2000; Ferrand & New, 2003; Jared & Seidenberg, 1990; Weekes, 1997) although there are some exceptions to this pattern, such as additive effects of frequency and consistency (Jared, 1997, 2002; Ziegler, Perry, & Coltheart, 2003). Yap and Balota found that all of the interactions were significant and all were in the expected direction (i.e., the effects diminished at higher frequencies). These interactions provide a highly constraining test for computational models since to simulate them, models need to capture the covariation between different variables rather than just the behavior of single variables by themselves.

### 1.1.3. Syllable number

One effect that is specific to reading aloud multisyllabic words is the effect of syllable number (i.e., the more syllables a word has, the longer it takes to read aloud, everything else being equal). Jared and Seidenberg (1990, Experiment 3) reported a significant effect of syllable number (see also Butler & Hains, 1979) and a significant interaction between syllable number and word frequency. Ferrand (2000) replicated the syllable number by frequency interaction in French. Recently, Yap and Balota (2009) also observed such an interaction in their ELP analyses, even after controlling for letter length, number of phonemes, frequency, and orthographic (e.g., Coltheart, Davelaar, Jonasson, & Besner, 1977) and phonological (e.g., Yates, 2005) neighborhood. Interestingly, effects of syllable number but not letter length were also present in their analyses of lexical decision latencies. This finding suggests that the effect of syllable number may not simply reflect phonological output processes, which gives some support to the claim of Álvarez, Carreiras, and Taft (2001) that “any model of lexical access has to incorporate a syllable level of representation or include the syllable as a sublexical unit in processing” (p. 553).

### 1.1.4. Consistency effects

The issue of how the consistency of the spelling-to-sound mapping affects reading aloud has been one of the primary areas of research not only for monosyllables but also for disyllables (Chateau & Jared, 2003; Jared & Seidenberg, 1990). Unlike frequency, length, and many other variables, consistency measures need to be redefined in the context of disyllabic words. In particular, consistency needs to be calculated for both the first and the second syllable.

Actually how consistency should be defined is not necessarily straightforward. One of the first studies to examine this question with disyllabic words was by Jared and Seidenberg (1990). In their study, they used a regularity metric based on whether a word contained a syllable with a spelling-sound correspondence that would be atypical when compared to the same syllable read aloud in isolation. They also used an all-or-nothing consistency measure, whereby inconsistent syllables were defined based on there being other words with the same orthographic syllable, but where the syllable in at least one of the other words had an exceptional correspondence in it. Their results showed effects of both of these measures in both the first and last syllables of words, with the effect being mainly evident in low-frequency words.

In more recent studies, consistency has been defined as a continuous measure for various spelling-sound relationships (e.g., Chateau & Jared, 2003; Treiman et al., 1995; Yap & Balota, 2009). Chateau and Jared (2003) evaluated a large number of these different types of relationships on a database of six letter words. These included both within-syllable and across-syllable mappings.

The within-syllable relationships examined by Chateau and Jared (2003) included consonants by themselves, the vowel by itself, the onset-vowel and vowel-coda (body-rime) relationship. Whilst the results they found were quite complex, they did find that the consistency of the second vowel was generally a good predictor of naming latencies. Yap and Balota (2009) also examined consistency with both onset and body-rime measures, and included both feedforward and feedback consistency, the second of which measures how consistently a word's pronunciation maps to its spelling (e.g., Ziegler, Stone, & Jacobs, 1997). All of their measures predicted some variance in the naming latencies of a large database of items they used, although, unlike the Chateau and Jared study, their results did not take into account the effect of vowel consistency alone.

Apart from within-syllable measures, both Chateau and Jared (2003) and Yap and Balota (2009) examined across-syllable measures too. The across-syllable measure Chateau and Jared examined was based on Taft's (1979) *Basic-Orthographic-Syllable-Structure* (BOSS). This metric splits disyllables into two representational parts (*orthographic syllables*) based on the maximization of consonant letters. The first part of the word that occurs before the split, excluding onset consonants, is known as the *Body of the BOSS* (i.e., the BOB). In its simplest definition (Taft, 1979), consonant maximization works by including all of the consonants that could legitimately occur at the end of a word in the first syllable as long as they do not break a morphological boundary (e.g., the BOB of *cradle* would be -ad, since -dl never occurs at the end of a word, but -d does). This can lead to cases where a grapheme<sup>1</sup> that corresponds to a phoneme in the second syllable is placed in the first orthographic syllable. Chateau and Jared found that BOB consistency was a comparatively strong predictor of naming latencies compared to the other metrics they examined, such as body-rime consistency.

Another measure of across-syllable consistency was examined by Yap and Balota (2009). This metric was based on Yarkoni, Balota, and Yap's (2008) idea that Levenshtein distance, a measure that allows for the graded inclusion of insertions and deletions of letters and phonemes into a similarity calculation, could be used to determine the similarity that orthographic and phonological patterns have with each other. Thus, unlike a simple positionally constrained metric, which predicts, for example, that *flog* and *log* are entirely different because there is no letter-letter overlap (i.e., the first letter in *flog* is different to the first letter in *log*, the second letter in *flog* is different to the second letter in *log*, etc.), this measure predicts that they have some similarity. Yap and Balota showed that a consistency measure based on a Levenshtein distance metric predicted a small amount of unique variance over and above just within-syllable measures, as did more simple measures that simply took the average value of within-syllable consistency metrics across syllables.

A number of studies have investigated the effects of consistency with disyllables in small-scale experiments (e.g., Chateau & Jared, 2003; Chen & Vaid, 2007; Jared & Seidenberg, 1990; Taft, 1979,

<sup>1</sup> The term grapheme can be used in multiple ways. Here, we use it to mean a single letter or a group of letters that is associated with a phoneme, but whose orthographic representation can be used independently from that association. For example, the graphemes in a word like *folk* might be f-o-l-k, even though -l is not commonly associated with any of the phonemes in the word *folk* (-l only sporadically occurs with words with /k/ in them). At present, the graphemes used in CDP++ represent a hypothesis about those people use and were selected by hand.

2001). In general, these studies have found that the consistency of various units facilitates reading aloud, although the results have typically been less reliable than those found with monosyllables in comparable manipulations, where extremely strong effects have often been found (e.g., Jared, 2002; Rastle & Coltheart, 1999).

### 1.1.5. Stress regularity<sup>2</sup>

A major challenge for models of multisyllabic word reading is the assignment of word stress. For some languages, such as French, this is not a problem, because French simply does not have lexically assigned word-level stress (e.g., Dupoux, Pallier, Sebastian-Galles, & Mehler, 1997). For English, however, word-level stress assignment is quite variable, and stress can fall on different syllables in different words. Despite this, there is a tendency to assign stress to the first syllable of disyllabic words, with 78% of such words being stressed on the first syllable (Ševa, Monaghan, & Arciuli, 2009). Because of this pattern, it has been suggested that disyllabic words in English can be considered *stress regular* if they are stressed on the first syllable and *stress irregular* if they are stressed on the final syllable (e.g., Brown, Lupker, & Colombo, 1994; Colombo, 1992; Monsell, Doyle, & Haggard, 1989).

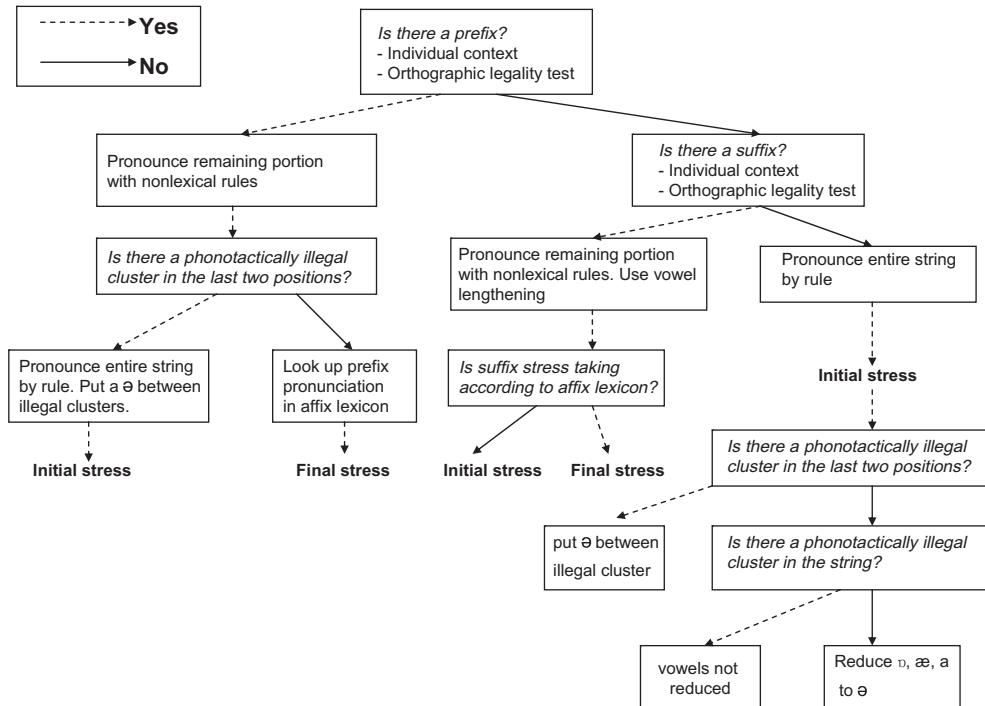
Monsell et al. (1989) were the first to investigate the effect of stress regularity and its possible interaction with word frequency in disyllabic English words. They found that although stress irregular low-frequency words were named more slowly than stress regular low-frequency words, neither the main effect of stress regularity nor the interaction between stress regularity and frequency reached significance (for a re-analysis of their data, see Rastle & Coltheart, 2000). Brown et al. (1994) repeated their experiment and found a main effect of stress regularity and an interaction between frequency and regularity that was close to significance. However, Rastle and Coltheart raised doubts about the validity of those results because neither item analyses nor item data were provided.

Because of the problems identified in the Brown et al. (1994) study, Rastle and Coltheart (2000, Experiment 1) attempted to produce a frequency by regularity interaction with a new set of English disyllabic words. As before, stress regularity was simply defined by default, that is, words with first syllable stress were considered regular and words with second syllable stress were considered irregular. The results, for both latency and error data, showed neither an effect of stress regularity nor an interaction between stress regularity and frequency. Because of these results, Rastle and Coltheart developed a more complex definition of stress regularity that was based on a rule-based system for stress assignment. This rule-based system was largely inspired by linguistic analyses of stress patterns in English by Fudge (1984) and Garde (1968), which suggested that 51 word beginnings and 101 word endings (most of which were morphemes) could predict the placement of stress. This system was implemented in the form of an algorithm that used the correspondences between these morphemes and the stress pattern typically associated with them to predict stress (see Fig. 1).

When stress regularity was defined on the basis of this algorithm (Fig. 1), Rastle and Coltheart (2000, Experiment 3) reported an interaction between frequency and stress regularity both in the latency and error data. The interaction was, however, only marginal in the latency analysis. Two additional studies that also used more complex definitions of stress regularity than a simple first syllable default also reported stress-regularity effects for words, but in both studies the effects were limited to errors and no reliable stress-regularity effect was obtained on RTs (Arciuli & Cupples, 2006; Kelly, Morris, & Verrekia, 1998).

Apart from small-scale experiments, the effect of simple stress position has also been examined in two large databases (Chateau & Jared, 2003; Yap & Balota, 2009). A similar result was found in both databases. In terms of RTs, there was either no significant effect (Chateau & Jared, 2003) or a very weak effect that was potentially caused by intercorrelated variables rather than by stress itself (Yap & Balota, 2009). Alternatively, in terms of errors, unlike the results of Rastle and Coltheart's (2000) first experiment, there appeared to be a small but significant effect, with words with first syllable stress being named more accurately than words with second syllable stress.

<sup>2</sup> Some studies use *typicality* and some studies use *regularity* to describe when a word is stressed in a way that is not as predicted for some reason. Here we use the term *regularity* even when the other studies have used *typicality*.



**Fig. 1.** Nonlexical stress assignment algorithm proposed by Rastle and Coltheart (2000).

Overall, the findings of both small-scale experiments (e.g., Arciuli & Cupples, 2006; Kelly et al., 1998) and large-scale database analyses (Chateau & Jared, 2003; Yap & Balota, 2009) converge to suggest that when stress is defined as a simple first syllable default, an effect of stress irregularity can only be found on errors. In contrast, stress effects on RTs were limited to a single experiment (Rastle & Coltheart, 2000, Experiment 3) that used a complex definition of stress regularity. We therefore suggest that stress-regularity effects on errors are the critical benchmark effect for models of reading aloud disyllabic words. To what extent stress effects can be obtained on RTs needs further investigation.

#### 1.1.6. Nonword reading

While stress assignment for real words could potentially be solved via a lexical lookup procedure (which of course would not predict the existence of a stress-regularity effect), people typically assign stress when reading nonwords such as *zortess* and *commoke*. Thus, a model must be able to assign stress nonlexically. This problem has been tackled by Rastle and Coltheart (2000), who developed the algorithm for stress assignment presented in Fig. 1. This algorithm can be applied to both words and nonwords. Rastle and Coltheart (Experiment 2) actually tested whether their algorithm would predict human stress assignment on a set of 210 disyllabic nonwords that were predicted by their algorithm to have either first syllable or second syllable stress. They showed that the algorithm agreed with the dominant stress pattern given by participants around 84% of the time. However, as noted by Ševa et al. (2009), their set of nonwords was somewhat biased towards good performance of the algorithm due to the majority containing affixes that were present in the morpheme list used by the algorithm to help predict stress. They therefore examined the performance of the algorithm on the more unbiased set of nonwords that was used by Kelly (2004) in a stress judgment task,<sup>3</sup> where stress was

<sup>3</sup> Whilst the study is not technically a naming task, it is likely to give very similar results.

found to be attracted by onset cluster complexity, a factor unrelated to the information specified in Rastle and Coltheart's algorithm. On those nonwords, the algorithm did not perform as well.

The complexity of consonant clusters in disyllabic nonwords has also been found to influence pronunciation. In particular, [Waese and Jared \(2006\)](#) examined how, in disyllabic nonwords, the length of the first vowel is influenced by the number of following consonants that occur before the second vowel. Waese and Jared compared three groups of nonwords. In one of the groups, a single consonant followed a single letter vowel (e.g., *bafest*), whereas in the other two groups two consonants followed. Of the groups with two consonants, one had consonant sequences that formed a legal second syllable onset (e.g., *baflet*) whereas the other had consonant sequences that did not form a legal second syllable onset (e.g., *bafnor*). The results showed that in the single consonant group, people were less likely to give short vowel responses (Single: 73%; Legal: 87%; Illegal: 93%) than the other two groups. Clearly then, the number of consonants that come after a vowel affects whether people are likely to produce the vowel long or short.

## 1.2. Previous models

### 1.2.1. Multi-trace memory model

The most well known model of reading aloud that deals with multisyllabic words is the connectionist multiple-trace memory model (MTMM) of [Ans, Carbonnel, and Valdois \(1998\)](#), which is currently set up to read French. In that model, all words are learnt in a connectionist network, and the same network is also used to read aloud nonwords. The network is structured such that there is an orthographic layer, where letters are input, and a phonological layer, where phonemes are output. Inputs into the orthographic layer use letter specific coding where the rightmost letter of the first vowel grapheme is centered at a “focal” point, and the rest of the letters occur in a contiguous sequence to the right or left of that letter. Representations in the phonological layer are centered on the vowel, with the other phonemes clustered around it in the same way letters are clustered around the focal point. When the model processes multiple syllables at the same time, the phonological layer is organized such that each syllable is coded separately around a vowel, unlike the orthographic layer, where letters are only ever clustered around a single focal point.

The MTMM has two ways of reading aloud, a *global* mode and an *analytical* mode. In global mode, all letters of the word or nonword are processed in parallel. In analytical mode, the word or nonword is decomposed into orthographic segments (generally syllables) and each syllable is read-out one-by-one by the model. The entire pronunciation is then built up based on multiple presentations.

Currently, the MTMM has no English implementation, which makes it impossible to test the model on the ELP database and the various benchmark effects described above. In addition, the model has no procedure to deal with stress assignment. This is not a problem for the original French implementation of the MTMM, because French has no word-level stress ([Dupoux et al., 1997](#)). However, it is a problem for implementing the model in English because it is not simply possible to change the database from French to English because how the model deals with stress would need to be specified. Therefore, a complete evaluation of the model has to be postponed until an English version of the model is made available.

### 1.2.2. The junction model

A second model of reading aloud that can deal with multiple syllables is that of [Kello \(2006\)](#), the Junction model. The Junction model uses simple recurrent networks (SRNs) that are trained to convert variable-length sequences into fixed-width representations and vice versa. For reading, an input SRN is used to encode letter sequences and phonemes into a fixed width representation, and another SRN is used to decode the fixed-width representations back. These representations and semantic ones are then bound together via a set of intermediate nodes, which allows the model to produce outputs (phonemes) from inputs (letters).

This “junction” at the intermediate layer, apart from being necessary to learn the input–output mapping, is theoretically important because it means that the model departs from the idea that there are two separate ways to get to phonology from print. This differs from most models, which converge on the assumption that phonology can be generated through a spelling–sound mapping process as

well as retrieved through a pathway that involves lexical (e.g., Coltheart et al., 2001; Zorzi et al., 1998a) or semantic (e.g., Harm & Seidenberg, 2004; Plaut et al., 1996) representations. Instead, in the Junction model, semantics, orthography and phonology are linked through a single, shared level of representation. In this sense, whilst it is still a connectionist model and still uses similar learning principles (i.e., backpropagation and its variants), it is theoretically very different from the Triangle framework initially proposed by Seidenberg and McClelland (1989). It is also very different from the two-layer network of CDP+ (Perry et al., 2007). CDP+ assumes that the relationship between orthography and phonology is generally very simple, and that the mapping is direct rather than mediated (Houghton & Zorzi, 2003; Perry et al., 2007; Zorzi et al., 1998a).

At present, the Junction model is in preliminary development. It was mainly tested on the ELP database (Kello, 2006; Yap & Balota, 2009), where it accounted for around 30% of the variance on the RTs of the words (Yap & Balota, 2009). A major problem it has, however, is poor nonword reading, where it produced errors on around 70% of the tested items (Kello, 2006).

A model similar to the Junction model has been recently proposed by Sibley, Kello, and Seidenberg (2010). This model includes stress nodes such that stress-regularity effects can be simulated. The model was also specifically designed to improve nonword reading, which was done by changing the input coding. Despite this, its error rate (15% with monosyllabic and 35% with disyllabic nonwords) remains very high in comparison to skilled readers. Moreover, nonword stress assignment (i.e., generalization performance) has not yet been tested.

### 1.2.3. Ševa et al.'s model of stress assignment

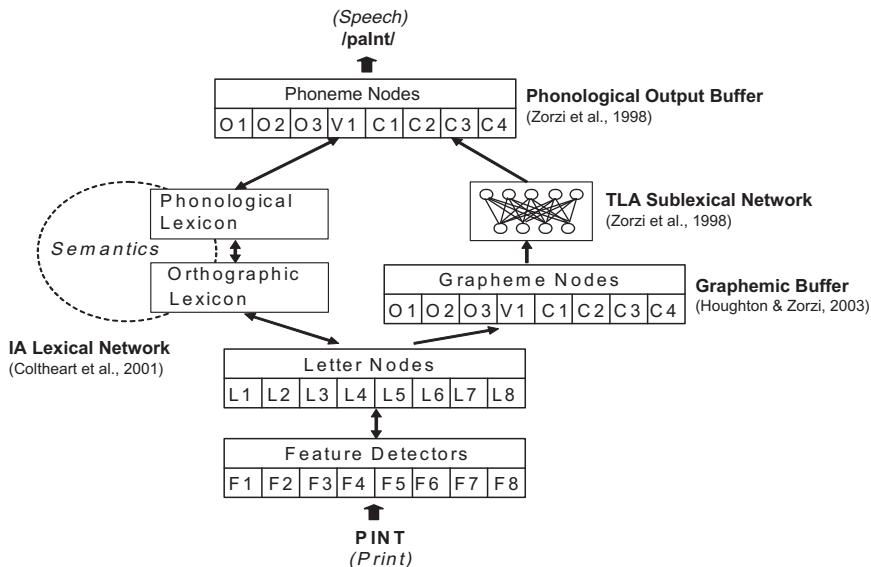
Within the connectionist framework, Ševa et al. (2009) developed a connectionist model of stress assignment for English to show that stress assignment is possible without using explicit linguistic rules (see also Gupta & Touretzky, 1994; Zevin & Joanisse, 2000). The model is a simple feedforward network that learns to map the orthography of words into stress position. The orthographic input layer is composed of 14 letter slots (364 input units). The input layer is fully connected to a layer of 100 hidden units, which in turn is fully connected to one output unit, which is used to represent which syllable is stressed.

On the CELEX database, the model of Ševa et al. (2009) learned to assign stress correctly for 97.0% of the words with first syllable stress and for 77.0% of the words with second syllable stress. In this respect, the model was slightly superior to the algorithm proposed by Rastle and Coltheart (2000), which obtained 92.5% and 75.6% correct classifications, respectively. The model was also tested on two nonword datasets (Kelly, 2004; Rastle & Coltheart, 2000) by examining how well it would predict each item, where each item was dichotomized into a correct and incorrect category based on which syllable the majority of participants assigned stress to. On the Rastle and Coltheart (Experiment 2) nonword data, the model correctly assigned stress on 87.7% of the items with first syllable stress and 49.5% of the items with second syllable stress. The algorithm of Rastle and Coltheart was superior on this dataset, yielding 93.0% and 74.7% correct classifications, respectively. On the Kelly set, the model produced 88.6% and 42.2% correct classifications for nonwords with first syllable and nonwords with second syllable stress, respectively. On this nonword set, Ševa et al.'s model outperformed the Rastle and Coltheart algorithm, which produced 78.2% and 43.8% correct classifications, respectively. Ševa et al. suggested that these results showed that a model that learns simple statistical relationships between orthography and stress may provide a simpler and more parsimonious account of stress assignment than the algorithm of Rastle and Coltheart.

## 2. Beyond single syllables: CDP++

In this section, we describe the basic architecture and processing assumptions of CDP++. Because CDP++ is built based on its direct precursor, we start with a short description of CDP+ (Perry et al., 2007).

CDP+ contains a number of independent representational levels (see Fig. 2). They can be broken down into two main parts: (1) the sublexical part, which contains the graphemic buffer and the two-layer network of phonological assembly (TLA network); and (2) the lexical part, which contains



**Fig. 2.** The overall architecture of CDP+. Note: Numbers shown inside the various layers index slot positions, whereas letters indicate the type of representation (f = features, l = letter, o = onset, v = vowel, c = coda).

the orthographic and phonological lexicons. The two parts are connected at the letter level and the phonological output buffer. The sublexical part of the model generates pronunciations for letter strings regardless of their lexical status and it is crucial for decoding novel stimuli (i.e., nonwords), with the graphemic buffer being used to organize single letters into the limited set of graphemes that are used by the model. The lexical part of the model is used to retrieve word pronunciations based on whole-word representations (i.e., access “the mental lexicon”).

The distinguishing – and most crucial – component of CDP+ is the TLA network (Zorzi et al., 1998a; Zorzi, Houghton, & Butterworth, 1998b). This network learns the most reliable mappings between orthography and phonology through its exposure to words that are encoded as sequences of graphemes and phonemes. The use of graphemes rather than individual letters is based on the hypothesis that the reading system uses a graphemic buffer where orthographic information is structured into a graphosyllabic template, with the most frequent graphemes used as representational units (Houghton & Zorzi, 2003).

To encode graphemes and phonemes, the TLA network uses a CCCVCCCC structure in both its input (graphosyllabic) and output (phonological) representations. At the phonological level, the Cs and Vs represent phonemes. Thus, phonological representations are not a linear string of phonemes as they are in some models (e.g., Coltheart et al., 2001), but rather, are structured into their syllabic constituents. At the orthographic level, the Cs and Vs represent consonant and vowel graphemes, rather than just the single letters that were used in CDP. This means that multi-letter graphemes (e.g., TH, EA, etc.) are encoded by a single unit. Thus, whenever a letter string is presented to the model, graphemes are first identified by a *graphemic parser* (with complex graphemes being preferred over simpler ones whenever there is potential ambiguity) and then placed in their appropriate slot in the buffer. Onset graphemes are assigned to the first three slots of the template (from left to right), the vowel grapheme to the vowel slot, and the remaining coda graphemes to the four remaining slots. In training, phonemes are assigned to the phonological template in exactly the same way.

Orthographic and phonological representations in the TLA network make direct contact with each other through the network connections (i.e., there are no intermediate layers of hidden units) and the relationship between graphemes and phonemes is learnt via a simple learning algorithm known as the delta rule (Widrow & Hoff, 1960). This rule is formally equivalent to a classical conditioning

law (the Rescorla–Wagner rule; Sutton & Barto, 1981) and has been directly applied to human learning (see Siegel and Allan (1996) for review). After training, when a stimulus is processed by CDP+, the graphemic parser computes grapheme representations from the letters available at the level of the letter detectors and inserts them into the graphosyllabic template. Activation spreads to the phoneme units of the TLA network, generating a plausible sublexical phonological representation.

## 2.1. Overview of CDP++

Extending CDP+ computationally to disyllabic words requires a number of modifications. These are: (1) The number of slots for coding letter features, letters, and phonemes was extended from 8 to 16<sup>4</sup>; (2) The set of graphemes was extended and the schwa phoneme, which was not used in CDP+, was added (for the full list of graphemes, see Appendix A); (3) The original graphosyllabic and phonological templates used by the TLA network were duplicated so that a second syllable could be processed. Thus, instead of using a single syllable CCCVCCCC graphemic and phonemic template for learning the relationship between spelling and sound, a disyllabic CCCVCCCC.CCCVCCCC template is used. Within each of the 16 possible grapheme and phoneme slots, all possible graphemes and phonemes may occur, although only onset graphemes are put in the onset slots of the first syllable and only coda graphemes are put in the coda slots of the second syllable; (4) The size of the lexicons were increased; and finally (5) the model was augmented with two sets of stress nodes that represent the position of stress, one set for sublexical stress assignment and one set for stress output. The complete architecture of CDP++ is presented in Fig. 3.

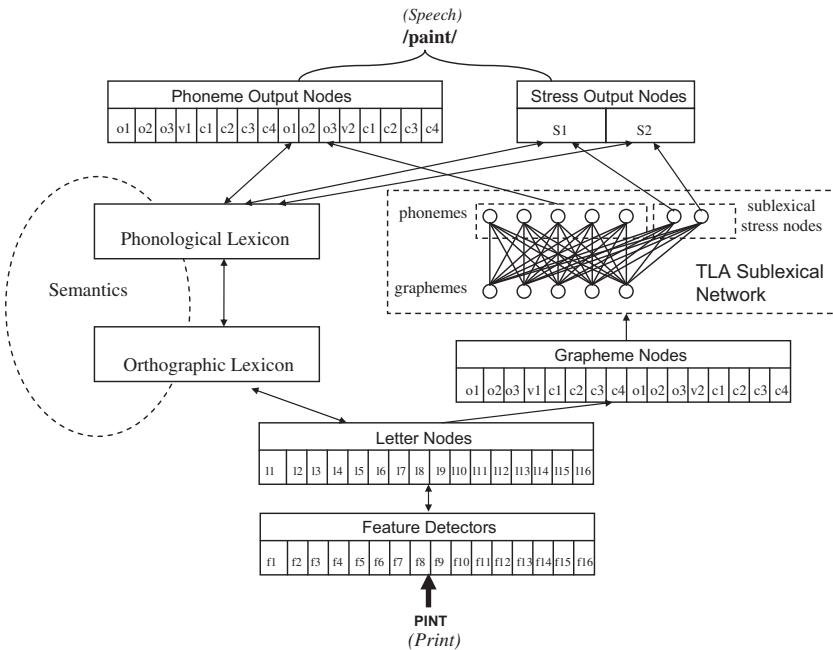
### 2.1.1. Learning spelling–sound relationships

As with CDP+, there is a distinction between training and running mode. The training phase is instrumental to learning the spelling–sound mappings in the TLA network. In training, lexical phonology is always available during learning (as is typical of connectionist models of reading aloud) and is organized based on an onset–rime distinction, as it was in CDP, except that it can be organized into two syllables rather than just one. This idea that the phonology used when reading might be organized into onsets and rimes represents a relatively uncontroversial assumption (e.g., Goswami, 2002; Treiman & Zukowski, 1996; Ziegler & Goswami, 2005). In training, phonology can be used in a top-down fashion – that is, the input representation can be changed depending on the phonology of the word the model is trained on. Phonological information is therefore used both as a teaching signal (i.e., to compute the error term in supervised learning) and to align graphemes to the positions that best represent the phonemes in a word. Finding a reasonable alignment ensures an efficient learning of spelling–sound relationships, avoiding the “dispersion problem” (Plaut et al., 1996) that is an intrinsic problem of slot-based coding.

The idea that representations should be aligned to help reduce dispersion is not unique to CDP++. The model of Bullinaria (1997) also used an alignment procedure, but the alignment occurs in the opposite direction as CDP++. In the model of Bullinaria, the output is aligned based on the input, whereas in CDP++, it is the input that is aligned based on the output.

A second unique feature of CDP++ is that both the orthographic and phonological representations are organized into syllabic groupings, rather than just phonological ones, as is the case with the MTMM. This is especially important for reducing the dispersion problem with long words. For example, consider the word *talking* with graphemes t-a-l-k-i-ng and phonemes /tɔ:kɪŋ/. If there was no syllabic organization and the graphemes and phonemes were aligned in simple linear order, then the simple one-to-one correspondences, which are generally the most commonly occurring, would be t → /t/, a → /ɔ:/, l → /k/, k → /ɪ/, i → /ŋ/, with –ng not mapping to anything. Alternatively, with syllabic

<sup>4</sup> There are in fact no disyllabic words used in the database with 16 letters (there are two words of 13 letters), although it is technically possible to create nonwords, albeit strange looking ones, with 16 letters and only two syllables (e.g., *chrautchedroosted*). Sixteen letter slots were used for the sake of simplicity as duplicating the other representations in the model (i.e., the graphosyllabic and phonological templates) meant they also had 16 slots. We consider this an implementation detail, rather than something that makes any strong predictions about the number of letters people can process in parallel.



**Fig. 3.** The overall architecture of CDP++. Note: Numbers shown inside the various layers index slot positions, whereas letters indicate the type of representation (f = feature, l = letter, o = onset, v = vowel, c = coda). S1 = first syllable stress; S2 = second syllable stress.

boundaries, -k would be identified to map to the first onset (/k/) of the second syllable. This means that the second syllable would have only simple one-to-one mappings, and thus only phonemes in the first syllable would have relationships more complicated than a simple one-to-one mapping in them.

To work out the best alignment for CDP++, graphemes identified from the letter string are moved into their most optimal positions. These positions represent a hypothesis about those that children would use when learning. In addition, we assume that the lexical phonology of a word is externally supplied or it can be guessed via phonology generated sublexically and via other contextual cues. Share (1995) provides strong arguments for the theoretical position that the generation of phonology is important, and we consider that CDP++ falls into this theoretical framework. Note that even if the phonology of every new word cannot be generated, this is not necessarily a problem for sublexical learning in CDP++, because all it means is that a small number of exemplars would not contribute to the learning of the orthography–phonology relationship, and this would make very little difference to the overall performance of the sublexical route.

At present, the positions for graphemes are determined computationally. This is done before training begins, and the positions remain the same throughout training (i.e., all of the input and output patterns are pre-coded before being submitted to the model). What needs to be determined in training is the position of graphemes that occur before the second vowel but after the first. All other graphemes are simply placed in a contiguous sequence in their respective onset and coda positions. This means that if there is only one vowel, the placement of graphemes is identical to CDP+.

Graphemes that occur between vowels are placed by using knowledge of the number of graphemes and phonemes that are in a word and grapheme–phoneme frequency. The number of graphemes in a word is calculated by simply selecting the longest graphemes possible, starting from the first letter of a word (see the [Supporting on-line materials](#)), and grapheme–phoneme frequency is calculated by

taking all words where the number of graphemes is equal to the number of phonemes, and, for each correspondence in those words, summing the logs of the word frequencies that they occur in.<sup>5</sup>

In the simplest case, where the number of graphemes and phonemes in a word is identical, the grapheme structure is simply the same as the phonological structure. For other cases, where the number of graphemes and phonemes that occur between the two vowels differs, the graphemes are aligned simply by identifying the most common grapheme that maps to the phoneme in the first onset position of the second syllable, and graphemes are aligned into an onset-vowel-coda structure based on that, as they were in CDP+. For example, when a word like *chalking* is encountered (which has the two graphemes –l and –k between the two vowels, but only one phoneme /k/), identifying which sound corresponds to the start of the second syllable (/k/) and which grapheme most frequently maps to it (i.e., –k → /k/ and not –l → /k/) provides the minimal knowledge required to effectively align all graphemes. In a broader theoretical perspective, this means that, during learning, phonology constrains the organization of graphemes into a syllabically-structured orthographic representation. From a developmental perspective, the idea that orthography has some organization is more plausible than a purely visual code (Goswami & Ziegler, 2006; see also Taft (1979, 2001) for an alternative possible orthographic organization).

It is important to realize that moving graphemes into positions based on phonology reduces dispersion at the grapheme–phoneme level. This is because without such an internal organization, the same graphemes would map onto different phonemes depending on the syllable structure of the word. For example, the –p in *ripe* corresponds to a phoneme in the coda of the first syllable. Alternatively, the –p in *ripen* corresponds to the first phoneme in the onset of the second syllable. With graphemic organization, only one-to-one relationships are learnt, because the grapheme –p in *ripe* and *ripen* is put in a different position in the input, since it is aligned with the /p/ phoneme. Without such organization, mappings would be learnt from –p to phonemes in two different positions, even though, for individual words, this pattern does not occur.

### 2.1.2. Running mode

When the model is run to perform a naming task, graphemes need to be selected and aligned in the graphemic buffer in a purely bottom-up fashion. It is therefore assumed that the graphemic parser attempts to align graphemes in the sublexical route so that they *approximate* where graphemes would go if phonological information were available. Under some conditions (e.g., heterophonic homographs) this approximation will be incorrect. This means that CDP++ is clearly different to models where the input for both learning and running mode is always presented in the same order and models that make no assumptions about graphemes or slots at all (e.g., Kello, 2006).

Parsing begins as soon as letters become available to the parser from the letter level (i.e., the first letter reaches an activation level above a predefined level). These letters are presented to the model in absolute position, and the letters input to the parser consist of the most active letter in each absolute position (note that processing is thresholded, like in CDP+; see Ziegler, Perry, and Zorzi (2009) for discussion). One of the main assumptions of CDP++ is that this parsing process is carried out from left to right on the letter string, with the letters parsed on-line into graphemes, and the graphemes put into the graphemic buffer. Therefore, the system must try and approximate not only what the graphemes are but the places in the template they go into.

The alignment of graphemes in the graphemic buffer is straightforward in the case of monosyllabic words (and virtually identical to CDP+), but it is more complex in the case of disyllabic words. It is more complex with disyllabic words because there is ambiguity in assigning graphemes to the different slots, in particular with regard to whether consonant graphemes after the first vowel should be

<sup>5</sup> Because the graphemes are selected independently and based on being the longest possible grapheme that is selectable, it means that some graphemes are occasionally selected whose letters might more appropriately be split. For example, with the word *hothead*, a -th grapheme is used (i.e., h-o-th-e-a-d). However, based on phonological information that is available in training only, it may be more sensible to split this into –t and –h, because in the word *hothead*, -t appears to map to /t/ and -h appears to map to /h/, and these are very common relationships that are simple to identify. Adding more complicated strategies to deal with these cases would certainly be possible. However, for the sake of simplicity, splitting graphemes based on phonological information was not done.

segmented into the first or the second syllable. For example, a word like *rapid* could potentially be segmented as *ra.pid* or *rap.id*. In the first case, the grapheme -p would go in the onset of the second syllable, whereas in the second case, it would go into the coda of the first syllable.

One way the parser could work out how to assign potentially ambiguous graphemes would be to use phonological linguistic constraints (e.g., Hall, 2006) and apply them at the orthographic level. The idea of using orthographic constraints to approximate phonological ones has been suggested by Taft (2001), who assumed that they can be used to help delineate orthographic segments, and Rastle and Coltheart (2000), who used them to help determine where stress in words should be placed. Such constraints could be implemented in the graphemic parser of CDP++.

An alternative view, and the view implemented in CDP++, is that many of the linguistic constraints that can be used to segment words do not have to be explicitly represented. This means that, at present, of the numerous possible linguistic constraints, the only one used in CDP++ is *onset maximization*, which is a well known and well accepted constraint in phonology (e.g., Hall, 2006). This means that consonant graphemes occurring between two vowels are assigned to the onset positions of the second syllable (i.e., from the 9th grapheme slot onwards), whenever possible. Thus, with the word *rapid*, the model would maximize the -p, and hence use the syllabification *ra.pid*, which is the same segmentation as the speech representation. This also means that words which include the same set of graphemes as another word may have identical graphemes assigned to different places – that is, the places graphemes are put may differ in different words, even if some of the orthographic sequences are shared. For example, with a word like *ripe*, the orthographic coding would be r-\*\*\*-i-p-e-\*\*\*\*\*-\*\*\*\*\*, and with a word like *ripen*, the orthographic coding would be r-\*\*\*-i-\*\*\*\*\*-p-\*\*\*-e-n-\*\*\*.

A second principle used in graphemic parsing comes from internal network dynamics (i.e., what has been learnt in the TLA network). The idea here is that the statistical information captured by the network during training provides implicit constraints that can affect the operations of the graphemic parser (see also Ans et al. (1998), who use this as their main constraint for segmenting orthography). In particular, the parser is prevented from inserting graphemes into a slot where nothing has been learnt (note that this information is readily available from the strength of the connection weights in the TLA network).

### 2.1.3. The phonology used in CDP++

A number of assumptions are made about the phonology of words that CDP++ uses. Basically, it is assumed that the phonology given in CELEX is largely correct. The assumption of CDP++ is therefore that the phonology of a word that is used in the reading system is essentially the same as that which one hears. Whilst this assumption has been used by all of the main models of reading aloud, it is certainly not the only one that has been argued for, and many people claim that the underlying phonology of a word can be different in some way to the surface phonology that people produce and hear, and sometimes radically so (e.g., Burzio, 1994; Harris, 1994).

CDP++ also makes the assumption that there is no ambisyllabicity at the phonological level – that is, a phoneme cannot be shared across syllables. Thus, words like *banner* keep their CELEX coding (/bæ.nə/) versus the ambisyllabic one (/bæn.nə/). Whilst there has been a reasonable amount of debate over this topic, some more modern accounts of English syllables go against the idea that ambisyllabicity is needed in their description (Hall, 2002; Hammond, 1997; Jensen, 2000).

One major case where the use of surface rather than underlying phonology is important is with the schwa phoneme (for reviews see, e.g., Heselwood, 2007; van Oostendorp, 1998). Schwa in English is temporally the most shortened vowel (Heselwood, 2007) and also the most frequently occurring (Roach, 2000). In CELEX, it is only ever transcribed as a single sound, although it has been argued that more complex distinctions between different types of schwa might exist in at least some English dialects (e.g., Flemming & Johnson, 2007; Ladefoged, 2001).

In general, it seems fair to say that schwas occur in words for qualitatively different reasons, although the exact nature of these reasons is extensively debated. For instance, schwa may be used instead of another phoneme when the typical pronunciation is not given for various reasons. Heselwood (2009), for example, argues that this occurs with some words with word final schwas in them in Received Pronunciation (e.g., *father*), and that the schwa is actually a vocalized allophone

of /r/. Schwa may also occur as a vowel in its own right when inserted between phonemes that cannot occur together (e.g., Halle & Idsardi, 1997). Some versions of this idea use an extreme approach, and do not even represent schwas in words when the position they occur in can be predicted by phonemes that cannot co-occur together (e.g., Heselwood, 2007). In this case, they are simply inserted into speech late in the speech output process based on the detection of various sonority violations and patterns amongst phonemes. Thus, for example, the phonology of a word like *command* would be /kmænd/ and the schwa inserted between /k/ and /m/ when the word is output after the illegal within-syllable /km/ sequence was detected. Finally, it has been argued that schwa may exist in the representations of some words but that it gets deleted in some cases at the speech output level (see e.g., van Oostendorp, 1998). In this case, whilst one might not be able to hear a schwa, it does not preclude its possible existence in the representation of a word.

Given the complexity of how schwa is represented and used in English and the continuing debate over it, at present, CDP++ simply assumes that the reading system treats schwa as a separate phoneme in its own right.

## 2.2. Implementation of CDP++

### 2.2.1. Database and words used

There are two main sets of words used by CDP++, one for training and one in the lexicon of the model. In terms of the lexicon, there are 32,270 orthography–phonology word pairs, of which 8228 are monosyllabic and 24,042 are disyllabic. The training database contains 30,516 orthography–phonology word pairs, of which 7920 are monosyllabic and 22,596 are disyllabic. These items and their frequencies were extracted from the CELEX phonological word form database (Baayen et al., 1993). The exact procedure used to extract the disyllabic words is presented in the Supporting on-line materials. Across both the training words and those used in the lexicon of the model, some initial pre-processing on the phonology of the words was done, which is also described in the Supporting on-line materials.

### 2.2.2. Lexical route

The lexical route was identical to that of CDP+ except that the number of letter slots was increased from 8 to 16 and the phonemes were changed to all of those used with disyllabic words (including the schwa phoneme). A single orthographic entry was used for heterophonic homographs, and the frequency of the orthographic entry was set to the summed frequency of the different words that the entry represented. Word frequencies of homophones in the phonological lexicon were also added together. After collapsing homophones and heterophonic homographs into single lexical items, the orthographic lexicon contained 31,873 and the phonological lexicon contained 29,841 unique entries.

### 2.2.3. Sublexical route

As in CDP+, the core component of the sublexical route is the TLA network that maps graphemes onto phonemes. As described above and as shown in Fig. 3, the template simply duplicated the graphemic buffer and phoneme nodes so that two syllables could be represented. Details on how the graphemes are assigned to the appropriate positions in the graphosyllabic template are provided below. Moreover, two sublexical stress nodes were added to the output layer of the TLA network (see “Stress system” below). The network is fully connected in the sense that all graphemes can potentially activate all phonemes and both stress nodes.

After training, some nodes in the network were assumed to be “dead”, and unavailable to the graphemic parser. A grapheme node was considered dead if the sum of all its weights (in absolute values) projecting to the phoneme layer was below a given constant (7.5 in the current parameters<sup>6</sup>), which

<sup>6</sup> Whilst it would have been convenient simply to set this value to zero, some graphemes occur very infrequently, and we wanted to stop the graphemic parser from using these, since they often contain very little information about spelling–sound correspondences because of their frequency and the strange spellings of the words they are generated from. For example, the word *isle* is coded using the graphemes i-s-l-e. This means there is in fact some learning between the letter -l and phonemes in the second coda slot of the first syllable. However, because this pattern occurs very rarely, the weights are not changed enough such that the sum of them is over the dead node level.

represents a value that almost all grapheme–phoneme relationships the model was trained on surpass. This means that most dead nodes occur when the connections originating from a grapheme node in a given position are not strengthened during training because the grapheme never appears in that position (e.g., a ‘j’ in the second coda position of the first syllable). When hitting a dead-node, the parser will be forced into a back-up strategy. Thus, nonwords like *jinje*, where the second –j would be in a dead-node position, are simple to identify (no learning would have taken place at the –j position) and readable by the model (see below for one example of such a strategy).

#### 2.2.4. Stress system

Stress is represented in CDP++ at two different levels. First, two *sublexical stress nodes* (see Fig. 3) represent predictions of the sublexical network about whether stress should fall on the first or second syllable (Fig. 3). These nodes are independent from the phoneme nodes although otherwise identical. The graphemes in the sublexical network and the sublexical stress nodes are fully connected. This means that when learning occurs, the model not only learns relationships between graphemes and phonemes, but also between graphemes and the stress nodes. Thus, in terms of learning, there is no difference between the way stress is learned and the way grapheme–phoneme mappings are learned. Indeed, identical training parameters are used for both. Information about stress is provided to the network during training by turning the appropriate stress node on or off, as determined by information from the database.

The sublexical stress nodes send activation to two *stress output nodes* that are placed at the level of the phonological output buffer. The stress output nodes also receive activation from the phonological lexicon, which provides information about lexical stress. Thus, the stress output nodes pool lexical and sublexical sources of stress information in much the same way that nodes in the phoneme output buffer combine phonological activation from sublexical and lexical routes.

The influence of stress on the naming process is governed by a parameter called the *stress node naming criterion*. Words are not read-out unless the activation of one of the stress output nodes has reached the criterion, even if the phonology is otherwise ready. This may therefore affect naming latencies in cases where the phonology of a word is ready to be read-out but neither of the stress nodes has reached the level of activation specified by the stress node naming criterion. Unless otherwise stated, the stress node naming criterion is set to 0.1 by default (see *General Discussion* for arguments as to why such a criterion is necessary).

In addition to the stress node naming criterion, the stress system uses four parameters: (1 and 2) an excitation and an inhibition parameter from the lexical route to the stress output nodes; (3) an excitation parameter from the sublexical network, which is the same parameter as the one which specifies the amount of activation that flows from the sublexical network to the phoneme output buffer; and (4) a lateral inhibition parameter, where activation from one stress output node may laterally inhibit the other. The parameter values appear in [Appendix B](#). The stress output nodes use the standard interactive-activation dynamics that the rest of the network uses. However, the activation from the sublexical network does not begin to activate the stress output nodes until the last letter in a word is processed by the graphemic parser. Thus, it is assumed that once the parser has processed the last letter, and hence when parsing comes to an end, the processing of sublexical stress information is triggered and begins. This choice is motivated by the fact that graphemes in the first syllable invariably tend to activate the first but not the second syllable sublexical stress node. It is not until the second syllable graphemes become available that the sublexical route can accurately assign stress.

#### 2.2.5. Training

The TLA network first received a phonics pre-training as in CDP+ (see [Hutzler, Ziegler, Perry, Wimmer, and Zorzi \(2004\)](#) for an extensive discussion of this approach, and the [Supplementary materials](#) for the list of pre-training exemplars). The phonics pre-training set consists of simple spelling-sound phoneme correspondences that are presented to the model (the complete list of correspondences is presented in the [Supporting on-line materials](#)). All of the correspondences occur in the first syllable, and represent less information than that which is given in many phonics training programs (e.g., [Lloyd & Wenham, 2000](#)). After the initial pre-training for 100 epochs, the network was trained on the training corpus for 40 epochs. This means that across 40 training cycles,

1220,640 ( $40 \times 30,516$ ) words are presented to the network. The order of the words in the training corpus was randomized. Training parameters used were identical to those in CDP+ and the learning rate (0.05) was scaled by the normalized log frequency of the trained word (i.e.,  $\log(\text{word frequency} + 2)/\log(\text{maximum word frequency} + 2)$ ); note that plus 2 is necessary because some words are of zero frequency. All weights start at zero and are updated after the presentation of each individual word.

**Graphemes used.** Graphemes used by CDP++ are identified by parsing the letter string from left to right. This is done based on the graphemes that CDP++ uses, which represent a hypothesis about those that people use, and lexical phonology does not influence this. When ambiguities about which graphemes to use occur, the longest grapheme in CDP++'s graphemic buffer is simply chosen. For example, with the word *might*, there is potential ambiguity in the -ght coda. It could be a single grapheme (-ght) or it could be two graphemes (-gh and -t). However, since -ght is not a part of CDP++'s grapheme set, this means that -gh and -t will be used. If -ght existed as a grapheme, then because it has more letters than -gh, it would have been used instead.

### 2.2.6. Graphemes alignment in training model

The alignment between graphemes and phonemes is important to optimize learning of the spelling-sound correspondences and maximize generalization performance (Perry et al., 2007; Plaut et al., 1996). Graphemes are assigned to the various positions in the graphosyllabic template according to what we call the 1-to-1 principle. The idea of this is that if there is a phoneme in a given position in the phonosyllabic template, there should be a grapheme in an identical position in the graphosyllabic template. A more complex alignment procedure is used when the 1-to-1 principle is violated and there is therefore ambiguity as to which graphemes belong to the first and second syllable of the template. This is solved by choosing the grapheme that co-occurs most commonly with the onset phoneme of the second syllable to start the new syllable. See the [Supporting on-line materials](#) for further details.

### 2.2.7. Grapheme parsing in running mode

Grapheme parsing in CDP++, like its predecessor CDP+, takes place using an attentional window which spans across three adjacent letters and moves from left to right over the string. Larger (i.e., multi-letter) graphemes are simply selected over smaller ones when there is a conflict. Graphemes are identified purely on the basis of orthographic information available from the letter level. Once the graphemes are identified, they are assigned to the appropriate slots in the graphemic buffer. At present, graphemes are identified from any letters that happen to be in the attention window. Thus, for example, if a single letter grapheme is identified and two letters are therefore left in the window, the parser will also try and identify what grapheme the two letters correspond to. If, as more letters become available, a longer grapheme can be used than those identified in the two letters that are left, the old grapheme will be replaced with the new one.

One crucial piece of information used in grapheme parsing is the number of vowel graphemes in the letter string (see e.g., Lupker, Perea, and Davis (2008) and Perea and Lupker (2004) for evidence that consonant-vowel identity is available very early in processing). If two vowel graphemes have been identified, then CDP++ assumes that it is processing a disyllable. There is one exception to this, which is the letter -e. In some cases, the purpose of the letter -e is similar to the other vowel graphemes – that is, it is generally used to signify that a vowel phoneme should be produced. However, the letter -e also has the purpose of making a short vowel long (cf. *bit* vs. *bite*). In this case, it needs to be recognized as something that should occur in the coda (as in CDP+ and also Plaut et al., 1996), rather than create a syllable in its own right. We assume that the decision on whether the -e is processed in the first or second syllable is a probabilistic judgment based on letter level information.

At present, differing forms of the letter -e are chosen by using a simple two-layer network that predicts whether the -e should be a coda -e or a vowel -e. Predictions are learnt by training the network on all words that contain -e as the second vowel grapheme. [Appendix C](#) illustrates how this is done with CDP++. Note that there are number of possible ways that this judgment could actually be made (e.g., purely graphemic information, a combination of both letters and graphemes, or purely letter information), and there is no data suggesting which way people might actually use, and so we consider this aspect of the model as tentative, and simply assume that people can make a probabilistic

judgment about the letter –e, with this being a demonstration that the information to do this exists within the database.

When the model is processing a monosyllable (i.e., if only one vowel grapheme has been encountered), CDP++ generally behaves as CDP+ does. However, when a grapheme cannot be assigned to a certain slot because the corresponding node does not exist (due to a dead node), it is simply inserted in the first onset position of the second syllable. For example, with the nonword *fanj*, the -j cannot be processed in the second coda slot, because no relationship is learnt in that position. Thus, the grapheme is moved to the onset of the second syllable, where this relationship is learnt (e.g., *banjo*). This is only done with coda consonants of the first syllable, although, in principle, a similar scheme could be done with vowels and onset consonants. Note that this is considered a simple “back-up” strategy for when the model encounters letter strings with extremely rare correspondences.

When the model is processing a disyllable (i.e., the second vowel grapheme forms a second syllable), the core constraint used in graphemic parsing is to *maximize the onset graphemes*. This is done by placing consonant graphemes after the first vowel in all available onset positions of the second syllable. There are two cases in which maximization of onsets will not occur for every grapheme. First, if there are more than three consonant graphemes; second, if a consonant grapheme cannot be assigned to the onset positions because a grapheme node is not available (i.e., there is a dead node). In these cases the assignment is revised by (a) shifting the leftmost consonant back into the coda of the first syllable and (b) shifting all of the graphemes to the right of the leftmost consonant one place back in the onset positions. This revision can be repeated, if necessary, until all graphemes occupy positions where nodes are available or there are no graphemes left in the onset positions.

An example of how onset consonants are maximized in disyllabic words can be seen with a non-word like *banvil*. With this nonword, there are two consonant graphemes between two vowel graphemes, and they could potentially be inserted into the onset positions. However, the –v grapheme cannot go into the second onset position of the second syllable – this relationship is simply not learned well enough by the model in that position and therefore the node cannot be accessed by the parser (i.e., there is dead node in that position). Therefore, the first –n grapheme is inserted into the coda of the first syllable, and the –v grapheme is assigned to the first slot of the second syllable. As can be seen, in this case, the intervocalic consonants are put in the places that will cause the correct segmentation, since the graphemic parser is sensitive to what is learnt by the TLA network, and generally it will not have learnt grapheme sequences that cause illegal segmentations. The efficiency of this method across a number of different word types is evaluated in the [Supporting on-line materials](#).

Finally, it is worth noting that grapheme parsing might fail in the case of orthographically illegal strings (e.g., *xskdol*), and more complicated backup strategies would need to be used. Such words could be identified because they violate orthographic constraints, and they could also be identified when the parser tries to assign a grapheme to a node where no learning has ever occurred (i.e., a dead node). It is important to note that whilst people can generate a response for these strings, [Ziegler, Besson, Jacobs, Nazir, and Carr \(1997\)](#) showed that such illegal letter strings are processed in a qualitatively different way than legal nonwords. As discussed in [Perry et al. \(2007\)](#), one way the model could handle these stimuli would be to use a grapheme-by-grapheme read-out strategy, where graphemes are placed in positions where information between them and phonology has been learnt.

#### 2.2.8. Parameters and parameter setting

The full list of parameters is reported in [Appendix B](#). The parameters were chosen in the same way as for CDP+ ([Perry et al., 2007](#)). The main difference between the parameter set of the two models is that the grapheme parsing speed in the sublexical route was increased in CDP++ from 15 to 10 cycles per letter. Increasing the speed much beyond this causes the quantitative performance of the model to decline on datasets where length effects are very important, such as [Weekes \(1997\)](#).

In CDP+, a vowel phoneme in the phoneme output buffer was always selected regardless of the activation level when the word was named. Accordingly, we used the same criterion in CDP++ for the first syllable. For the second syllable, the most active vowel was always chosen even if it was not above the naming threshold if a coda consonant was active in the second syllable. This was done because if a coda consonant is activated in the second syllable, it means the syllable must always also contain a vowel.

### 2.2.9. Running the model

The model was always run with the full lexicon and with identical parameters, unless stated in the text.

## 3. Results

### 3.1. Overall performance

When all of the 32,270 words in the lexicon of the model were run with the standard parameter set, all but 285 (.88%) were given the correct phonology, and 264 had stress errors (.82%). The errors made by the model were not random, but rather fell into certain categories. Of the major categories, (a) 68% were pronunciations in which the model missed the final phoneme (e.g., saying *transcribe* for *transcribed*), and almost all of these were on long morphologically complex words. Inspection of the activation dynamics of these erroneous responses showed that the correct orthographic and phonological lexical entry typically reached ceiling (i.e., an activation of 1) and that the final phoneme was inhibited by the morphologically simpler neighbor being activated later in processing; (b) 14% were missing a phoneme. Inspection of the activation dynamics of these words showed that this was normally caused by two different phonemes of heterophonic homographs competing with each other, meaning that neither reached the phoneme naming activation criterion; (c) 2% were given monosyllabic responses even though the correct answer was disyllabic; and (d) 15% represented alternative pronunciations that could occur if the words were read like nonwords.

### 3.2. Monosyllabic word reading

The first critical test for a new model of disyllabic word reading is to check whether it can still account for the monosyllabic benchmark effects (e.g., consistency, regularity, length, etc.). In other words, whether the new model is backwards compatible needs to be checked. This is a nontrivial issue because the first syllable of the model is trained on both disyllabic and monosyllabic words. We therefore examined its performance on the list of monosyllabic benchmark effects proposed in Perry et al. (2007). These effects are summarized in Table 1. The results of the simulations as presented in Appendix D, and, as shown there, CDP++ was able to simulate all benchmark effects as well as CDP+ except for body neighborhood.

### 3.3. Database performance

To examine whether CDP++ performs as well as its predecessor (CDP+) on the influential monosyllabic databases, we examined the performance of CDP++ on the four main databases used in previous model tests (Balota & Spieler, 1998; Seidenberg & Waters, 1989; Spieler & Balota, 1997; Treiman et al., 1995). Note that Balota and Spieler used a population of older adults, which is not typical of all of the other studies examined. As can be seen from Table 2, CDP++ performs slightly better than CDP+ on all of these and hence at a higher level than the other models.<sup>7</sup>

Of course, the most critical test for the new model is how well it can deal with a disyllabic database. The best database available to examine this is the ELP database of Balota et al. (2007). That database has RTs to 22,144 monosyllabic and disyllabic items. Of these, 18,126 are in CDP++'s lexicon. All of these words were therefore run through CDP++. After the exclusion of 133 phonological errors (including three words that did not complete processing within a 300 cycle limit) and 152 RT outliers that were outside a three standard deviation (3SD) cutoff calculated from all words that were not phonological errors, there were 17,841 CDP++ responses. The model was also tested on two additional large-

<sup>7</sup> In Perry et al. (2007) we did not use a three standard deviation cutoff on the large databases. However, it was used here and was calculated from the reaction times of all phonologically correct responses that were given. This was done because a very small number of outlier items in some of the databases caused quite large changes in the  $R^2$  values. For example, leaving in the 28 outliers in the Spieler and Balota (1997) and Balota and Spieler (1998) database causes the amount of explained variance to drop from 19.5% and 24.0% to 12.3% and 16.0%. Of the outliers, 17 were heterophonic homographs and 10 were highly inconsistent.

**Table 1**

List of monosyllabic benchmark effects (from Perry et al. (2007)). Tick marks indicate successful simulations (for details, see Appendix D).

Name of effect	Description	CDP+	CDP++
Frequency	High-frequency words are faster/more accurate than low-frequency words	✓	✓
Lexicality	Words are faster/more accurate than pseudowords	✓	✓
Length × lexicality	Nonword naming latencies increase linearly with each additional letter	✓	✓
Frequency × regularity	Irregular words are slower/less accurate than regular words. This effect is typically larger for low-frequency words (Paap and Noel, 1991) but has also been reported for high-frequency words (Jared, 2002)	✓	✓
Word consistency	Inconsistent words are slower/less accurate than consistent words. The size of the effect depends on the friend–enemy ratio	✓	✓
Nonword consistency	Nonword pronunciations show graded consistency effects; that is, people do not always use the most common grapheme–phoneme correspondences	✓	✓
Position of irregularity	The size of the regularity effect is bigger for words with first position irregularities (e.g., <i>chef</i> ) than for words with second or third position irregularities	✓	✓
Body neighborhood	Words with many body neighbors are faster/more accurate than words with few body neighbors	✓	–
Pseudohomophone advantage	Nonwords that sound like real words (e.g., <i>bloo</i> ) are faster/more accurate than orthographic controls	✓	✓
Surface dyslexia	Patient MP showed a specific irregular word reading impairment that was modulated by the consistency ratio of the words as well as their frequency	✓	✓
Phonological dyslexia	Patient LB showed a specific nonword reading impairment which was reduced with pseudohomophones orthographically similar to their base words	✓	✓
Masked priming	Words preceded by a masked onset prime are faster/more accurate than words preceded by unrelated primes	✓	✓

**Table 2**

Percentage of variance accounted for ( $R^2$ ) by CDP++, CDP+ (Perry et al., 2007), CDP (Zorzi et al., 1998a), the Triangle model (Plaut et al., 1996), and the DRC (Coltheart et al., 2001) on the Spieler and Balota (SB, 1997), Balota and Spieler (BS, 1998), Treiman et al. (1995), and Seidenberg and Waters (SW, 1989) databases.

Database	Models				
	CDP++	CDP+	CDP	Triangle	DRC
SB (1997)	19.5	17.3	5.9	3.3	3.7
BS (1998)	24.0	21.6	6.7	2.9	5.5
Treiman	18.1	15.9	6.5	3.3	4.8
SW	10.9	9.6	2.7	3.0	6.1

scale item sets. These were the monomorphemic items selected from the ELP database by Yap and Balota (2009), for which it produced 6500 correct and usable responses (79 outliers, 13 phonological errors, 132 items not in CDP++'s lexicon, and three items that did not finish processing by 300 cycles, were excluded) and the database of Chateau and Jared (2003), for which it produced 866 correct and usable responses (8 outliers, 4 phonological errors, and 23 items not in CDP++'s lexicon were excluded). The model also made 91, 35, and 6 stress errors on the ELP, Yap and Balota, and Chateau and Jared items.<sup>8</sup> Of these, the vast majority were words with second syllable stress that were assigned first syllable stress (85%, 89%, and 100% for the three databases, respectively) despite words with second syllable stress being the minority in all of the databases. The model therefore tends to make most of its stress errors on stress irregular words (i.e., words with second syllable stress).

To examine the performance of CDP++ on these databases, we used a hierarchical regression analysis with two steps. In the first step, we included Yap and Balota's (2009) onset coding (i.e., surface

<sup>8</sup> Unlike the errors, time-outs, and outliers, the RTs of these words were left in the following database analyses.

level coding) in order to take into account systematic variation due to different phonetic onsets (see e.g., Kessler, Treiman, & Mullenix, 2002), which CDP++ does not capture. That scheme consists of 13 features used to describe the initial phoneme of words (affricative, alveolar, bilabial, dental, fricative, glottal, labiodentals, liquid, nasal, stop, velar, and voiced) which are coded as “on” if they exist in the onsets of the words. In the second step, the RTs of CDP++ were added. With these two steps, the regression provides an estimate of the overall performance of the model, taking into account onsets characteristics of the words.

Apart from simply using an overall measure of how well CDP++ predicts the data to evaluate the model, the percentage of variance explained by CDP++ can also be compared with the percentage of variance explained by the major word recognition variables, such as frequency, length, consistency etc. To examine this, we used the same type of analysis as Yap and Balota (2009), where variables of interest were examined using a multi-step hierarchical regression. In the first step, surface variables (onsets) were added. In the second step, frequency was added as it is the strongest correlating factor in the databases (Spieler & Balota, 1997) and is thus worthwhile examining by itself. In the third step, other common lexical variables were entered (orthographic neighbors, phonological neighbors, syllable number, orthographic length). These variables were the same as those used in Yap and Balota (2009). In the fourth and final step, different spelling–sound consistency measures were added.

Since different database analyses have used different variables and there is a vast number of possible variables that could potentially be examined, for the sake of simplicity we took variables that have been commonly used across a variety of studies. In particular, we used the onset coding of Yap and Balota (2009) in all analyses. Furthermore, we used log-transformed frequency rather than using rank-transformed frequency as Yap and Balota (2009) did because this is the most common way frequency has been examined in other databases. Finally, we used only two higher level measures, both of which were designed to examine spelling–sound consistency. One was composite consistency, which was either the average body–rime type consistency of both syllables in disyllabic words or the type consistency of the first and only syllable in monosyllabic words. Body–rime consistency was used as it has been used numerous times in both monosyllabic and disyllabic studies (e.g., Chateau & Jared, 2003; Spieler & Balota, 1997; Yap & Balota, 2009). Yap and Balota also found that this measure produced the standard Consistency by Frequency interaction in their data. The second phonological measure we used was BOB consistency. Whilst this measure is less commonly used than body–rime consistency, it was very important in the study of Chateau and Jared (2003), in which it predicted vowel pronunciations better than first syllable body–rime consistency.

All of the consistency metrics were calculated based on the words selected from the CELEX database that CDP++ was trained on. These were used rather than the full CELEX database because it is not always obvious in the full database how letter strings should be segmented. BOB consistency was calculated based on Taft's (2001) definition that the BOB division occurs at the end of a sequence of letters once a letter occurs that does not exist in an extant coda in extant monosyllables. We did not consider final letter –e as a special case (unlike Taft, 1979) and morphological structure was ignored. The actual BOB divisions of words were found using an automatic learning procedure and a small number of segmentations were entered by hand.

The results are presented in Table 3. On the Yap and Balota (2009) items, CDP++ performed very similarly to all of the predictors entered in the final step, that is, it accounted for 45.4% of the variance while all variables together explained 45.7%. On the ELP and Chateau and Jared (2003) items, it accounted for 36.9% and 33.8% of the variance, respectively. This performance is slightly below the variance accounted for by all variables together (i.e., 39.6% and 38.5%). Because reading aloud latencies are influenced by many factors outside the scope of the model, some of which may cause the distribution of results to differ to that which the model produces, we repeated the same analysis using ranked ordered human and model RTs. This caused the amount of variance explained by CDP++ to increase in all cases (see Table 3).

Although the quantitative performance of CDP++ appears quite high on the databases, the variance accounted for is still far from 100%. However, given the inherent noise and variability in these large-scale data sets, it is not clear what the maximum amount of variance that CDP++ should be able to account for is. One possible way to investigate this is to examine how well different databases correlate with each other and compare that to CDP++. This is possible because there is a large amount of

**Table 3**

Percentage of variance ( $R^2$ ) explained in different steps of the hierarchical regression analysis on the ELP (Balota et al., 2007), Yap and Balota (2009), and Chateau and Jared (2003) items and by CDP++ (including onset coding).

Predictor variables	ELP n = 17,960	Yap & Balota n = 6552	Chateau & Jared n = 875
<i>Step 1: surface variables (onsets)</i>	9.0	10.1	9.9
<i>Step 2: log word frequency</i>	29.8	32.2	29.8
<i>Step 3: standard lexical predictors</i>	36.7	43.0	33.4
<i>Step 4: consistency variables</i>	39.6	45.7	38.5
CDP++ and onset coding	36.9	45.4	33.8
CDP++ and onset coding (rank ordered)	40.8	49.4	40.3
CDP++ n	17,841	6500	861

Note: n values are slightly different between CDP++ and the databases as errors were excluded from the CDP++ items.

overlap in the items that were used across some of the databases. Correlations between shared items in the databases may therefore give some idea of how much variance can potentially be explained. One technical problem with examining the data with simple correlations is that the actual triggering of the voice keys may have differed across studies (see e.g., Kessler et al., 2002). This could have caused differences in results based on factors that are not due to cognitive process involved in reading that we are interested in. Therefore, rather than examine plain correlations, we examined  $r$  values where item RTs from one database plus the onset coding scheme used by Yap and Balota (2009) were used to predict the item RTs from another database. We also added onset coding to the responses from CDP++ when examining its performance on predicting results from the databases.

As can be seen in Table 4, the identical items across the four different databases correlated moderately, with  $r$  values ranging between .42 and .68 (i.e., 17.6% and 46.2% of the variance), with the percentage of variance accounted for by CDP++ being generally similar to the percentage that was obtained when using one database to predict another. A similar analysis examining the relationship between shared disyllabic words in the ELP (2007) and Chateau and Jared (2003) databases showed  $r$  values of .71 and .70 (i.e., around 50% and 49% of the variance when the ELP items plus onset coding were used to predict the Chateau and Jared items and vice versa). On these databases, CDP++ plus onset coding showed  $r$  values of .55 and .58 (30.3% and 33.8% of the variance). This difference shows that there is still some room for improvement in accounting for variance in disyllabic databases.

### 3.4. Syllable number effect

As noted in the Introduction, syllable number is an effect that is specific to the reading aloud of multisyllabic words. We examined whether CDP++ would show a syllable number effect over and

**Table 4**

$R$  values from regression analyses using onset coding plus item means shared across a number of databases (Spieler & Balota (SB), 1997; Balota & Spieler (BS), 1998, Treiman et al. (TR), 1995; Seidenberg and Waters (SW), 1989, and the ELP (2007)) and CDP++ to predict item means across the same databases.

Predicted item means	Onset coding + database of predictor items means					
	CDP++	SB	BS	TR	SW	ELP
SB	.54		.64	.47	.44	.56
BS	.59	.68		.52	.55	.64
TR	.44	.46	.44		.42	.47
SW	.66	.65	.67	.64		.68
ELP	.62	.63	.66	.58	.58	

$n = 1100$ . Note:  $r$  values differ above and below the diagonal because using onset coding plus the items from one database to predict the items in a second does not lead to an  $r$  value which is identical to using onset coding plus items from the second database to predict items in the first.

above the effect of letter length. The ELP dataset was therefore split into categories based on letter length (3–8 letters) and number of syllables (one or two). Because of the phonetic bias effects mentioned above, we first computed response time residuals, in which the effects of phonetic bias due to the onsets were removed via regression, and these were examined rather than raw scores. The results from the human data showed that disyllabic words were slower to process than monosyllabic ones, and that longer words were slower to read aloud than shorter ones. However, there was an interaction, where the difference between monosyllabic and disyllabic words decreased across word length. As can be seen in Fig. 4, CDP++ produces a very similar pattern as the human data. Note that the groups are inherently confounded on a number of different variables that differ across the categories, such as frequency. Reasons for this interaction are discussed in the General Discussion.

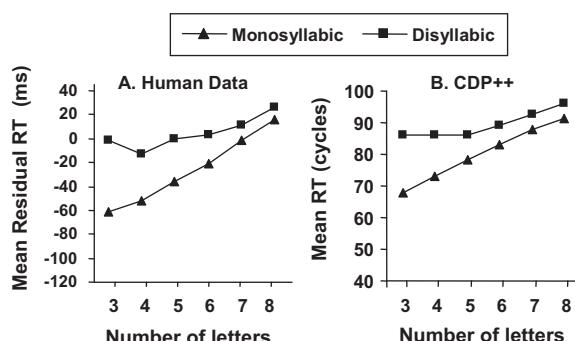
### 3.5. Consistency effects

As can be seen in Appendix D, CDP++ generally displays a pattern of consistency and regularity effects that is similar to that of people with monosyllables. Apart from monosyllables, there are two important studies examining consistency and regularity effects in disyllabic words: [Jared and Seidenberg \(1990\)](#) and [Chateau and Jared \(2003\)](#).

The [Jared and Seidenberg \(1990\)](#) study provides the simplest test of CDP++, as they examined both consistency (using regular-inconsistent words and controls) and regularity (using “exception” words and controls) in the first and second syllable positions of words. To examine how well CDP++ predicted the data, the by-items results were recalculated after the removal of 32 out of 160 of the words because they were either trisyllabic or not in CDP++’s lexicon (between 4 and 6 items in each group). One word was also removed from the controls because it was highly irregular with respect to CDP++’s database (*enlist*, which begins with an /ɪ/). The removal of the items did not affect the frequency matching across groups, as there were no significant differences between the regular-inconsistent/exception words and their controls, all  $p > .16$ . Note, however, that the means did change somewhat from the initial study and thus the present results need to be taken with some caution. The new means appear in [Table 5](#).

A re-analysis of the [Jared and Seidenberg \(1990\)](#) data with an ANOVA using consistency (inconsistent vs. consistent), consistency type (consistency vs. regularity) and place of inconsistency (first vs. second syllable) showed that words with some form of inconsistency (i.e., exception or inconsistent) were slower to name than their controls,  $F(1, 120) = 9.19$ ,  $MSE = 33,315$ ,  $p < .005$ . CDP++ showed a similar result,  $F(1, 116) = 5.01$ ,  $MSE = 400$ ,  $p < .05$ . No interactions were significant in either the human or model data. CDP++ also explained 22.0% of the individual item variance. Due to the item set being changed from that of [Jared and Seidenberg](#), further comparisons were not pursued.

The study of [Chateau and Jared \(2003\)](#) further explored consistency effects in disyllables using more sophisticated continuous metrics of consistency rather than the dichotomous classifications of [Jared and Seidenberg \(1990\)](#). Across a number of experiments, they examined BOB consistency, VC



**Fig. 4.** Mean human and CDP++ reaction times (RTs) of monosyllabic and disyllabic words on the full ELP (2007) database.

**Table 5**

Effects of consistency and regularity in disyllabic reading aloud. Results are from a re-analysis of the human data (in milliseconds) of Jared and Seidenberg (J&S, 1990). CDP++ simulations are in cycles.

	Consistency			Regularity		
	Incon.	Con.	Effect size	Irreg.	Con.	Effect size
<b>J&amp;S (1990) re-analysis</b>						
<i>Item means</i>						
First syllable	592	577	15	639	587	52
Last syllable	623	589	34	622	593	29
<i>CDP++</i>						
First syllable	85.8	83.7	2.1	93.1	85.2	7.9
Last syllable	88.5	85.4	3.1	89.1	87.7	1.4

Note: Incon. = inconsistent; con. = control; irreg. = irregular.

(i.e., body–rime) consistency in the first syllable and vowel consistency in the second syllable. The two experiments where BOB consistency was examined (Experiments 2 and 3), were marginally significant in the Chateau and Jared data, but not significant at all with CDP++,  $t < 1$ . With first syllable VC consistency, Chateau and Jared found no significant effect in the first syllable. With second syllable vowel consistency, a marginally significant difference was found. CDP++ displayed a similar pattern of VC and V consistency, with no significant effect of VC consistency found in the first syllable (High VC consistency: 86.1, Low VC consistency: 89.8,  $t < 1$ ), and a significant (and hence slightly too strong) effect of vowel consistency in the second,  $t(44) = 2.44$ ,  $SE = 2.70$ ,  $p < .05$  (High Consistency: 85.7; Low Consistency: 92.4).

### 3.6. The Yap and Balota test

Until now, most modelers have looked at quantitative fits between the models and human data by simply regressing the model latencies onto human latencies. More recently, however, [Yap and Balota \(2009\)](#) proposed a new test, henceforth referred to as the Yap and Balota test, which consists of regressing model and human latencies onto key word recognition and naming variables (frequency, consistency, length, etc.). This analysis allows one to test whether a model's latencies are influenced to the same extent by the variables that affect the human latencies, thus providing a much more fine-grained test for the quantitative fit of a model than the percentage of variance accounted for.

To examine CDP++ at this more fine-grained level, we computed regressions between model and human latencies and various common lexical variables on the full ELP database, the monomorphemic items used by [Yap and Balota \(2009\)](#), and the items from [Chateau and Jared \(2003\)](#). Whilst there are a large number of variables that could potentially be used in the regressions, we used the surface (onset descriptors) and the standard lexical variables of Yap and Balota, although some were calculated with slightly different metrics. These metrics and the steps used to examine the data were the same as those used above (see [Table 3](#)), except that instead of entering frequency alone in Step 1, it was entered together with other common lexical variables. The results are presented in [Table 6](#).

As can be seen from [Table 6](#), all of the variables examined were significant, and the directions of the  $\beta$ -coefficients with CDP++ were the same as the directions of the  $\beta$ -coefficients with the human data, excluding some of the neighborhood comparisons. The magnitudes of the effects were comparable apart from some systematic differences. First, almost no variance was explained by the surface variables with CDP++, whereas surface variables explained some variance in the human data. This is of course exactly what is expected, because these effects reflect the differential triggering of voice keys in response to different phonetic onsets, a process that is not implemented in CDP++. Second, the frequency effect was much stronger with CDP++ than the human data. This is probably due to the fact that the frequency counts in CDP++'s lexicons are exactly the same as those used in the regression (both use CELEX counts), thus potentially inflating the correlation. It is possible to examine this hypothesis by changing from CELEX frequency counts to another frequency count, such as the log Hyperspace Analog to Language (HAL) frequencies ([Lund & Burgess, 1996](#)). When this is done, the

**Table 6**

Standardized  $\beta$ -coefficients from hierarchical regression analyses predicting human and CDP++ data on items from the ELP (Balota et al., 2007), Yap and Balota (2009), and Chateau and Jared (2003).

Predictor variable	Item set					
	ELP <sup>a</sup>		Yap and Balota <sup>b</sup>		Chateau and Jared <sup>c</sup>	
	CDP++	RTs	CDP++	RTs	CDP++	RTs
Step1: surface variables ( $R^2$ )	.009	.090	.016	.101	.015	.099
Step 2: standard lexical variables						
Log frequency	-.60**	-.37**	-.62**	-.38**	-.77**	-.44**
Number of syllables	.11**	.073**	.17**	.14**		
Letter length	.33**	.13**	.26**	.22**		
Orthographic neighborhood	-.072**	-.16**	-.039**	-.070**	.005	-.11*
Phonological neighborhood	-.035**	-.044**	-.050**	.050**	-.19**	-.11*
$R^2$	.770	.367	.755	.430	.656	.334
$\Delta R^2$	.761	.277	.739	.329	.641	.235
Step3: consistency variables						
Average rime consistency	-.12**	-.11**	-.17**	-.12**	-.18**	-.18**
BOB consistency	-.032**	-.098**	-.050**	-.087**	-.066*	-.11**
$R^2$	.785	.396	.787	.457	.696	.385
$\Delta R^2$	.015	.029	.032	.026	.040	.051

Note: N values represent all words where metrics and responses exist on all measures.

\*\*  $p < .001$ .

\*  $p < .01$ .

<sup>a</sup>  $N = 17,960$  (human)/17,580 (CDP+).

<sup>b</sup>  $N = 6552$  (human)/6470 (CDP+).

<sup>c</sup>  $N = 875$  (human)/863(CDP+).

frequency betas for the model are much closer to those of the human data, although still slightly higher than the human ones (CDP++/Human: ELP:  $-.51/-42$ ; Yap and Balota:  $-.53/-41$ ; Chateau and Jared  $-.65/-41$ ).

There are two additional discrepancies between the human data and CDP++. First, the model always produced a smaller beta coefficient for BOB consistency than the human data. Thus, it is likely that the model is not as sensitive to this variable as people are. Second, the model produced facilitatory neighborhood effects for phonological neighbors on the [Yap and Balota \(2009\)](#) items, whereas the human data showed an inhibitory effect. Note, however, all other data sets showed facilitatory effects of phonological neighbors, which is also more consistent with other literature (e.g., [Mulatti, Reynolds, & Besner, 2006](#); [Yates, 2005](#)). The model also missed the facilitatory effect of orthographic neighbors in the [Chateau and Jared \(2003\)](#) data. Apart from these small discrepancies, the model correctly simulated all other neighborhood effects. Together then, although the model was not perfect in simulating all neighborhood effects, it correctly simulated the major pattern, which seems to be a facilitatory effect of orthographic and phonological neighbors.

Whilst the previous analysis examined the effect of various variables by themselves, [Yap and Balota \(2009\)](#) also suggested that there are theoretical reasons to believe that some interactions are also important. We therefore investigated the four two-way interactions they did, the Frequency by Length, Frequency by Orthographic Neighborhood, Frequency by Syllable Number, and Frequency by Consistency (both composite rime and BOB consistency) interactions. This was done in the same way as Yap and Balota, where surface and standard lexical variables were entered into a regression as well as the interaction term. Standard lexical variables were centered if they were part of the interaction term to reduce collinearity ([Aiken & West, 1991](#)) and consistency variables were only entered if they were also part of the interaction. The results appear in [Table 7](#).

As can be seen, all of the interactions were significant, excluding the Frequency by BOB interaction in the real data on the [Chateau and Jared \(2003\)](#) item set. This result, however, is probably due to a lack of power, because the beta value for the non-significant interaction was the same as the two other data sets which had many more items, thus allowing significance to be found even though the effect size was tiny.

**Table 7**

Standardized  $\beta$ -coefficients of critical two-way interactions predicting human and model data on items from the ELP (2007), Yap and Balota (2009), and Chateau and Jared (2003).

Interaction	Item set					
	ELP (2007)		Yap and Balota (2009)		Chateau and Jared (2003)	
	CDP++	Human	CDP++	Human	CDP++	Human
$F \times$ syllable number	−.084**	−.097**	−.049**	−.068**		
$F \times$ length	−.054**	−.086**	−.057**	−.076**		
$F \times$ consistency	.054**	.068**	.075**	.079**	.12**	.087*
$F \times$ BOB	.037**	.045**	.044**	.045**	.071**	.045
$F \times$ orthographic neighborhood	.086**	.12**	.095**	.17**	.066**	.12**

Note:  $F$  = frequency.

\*\*  $p < .001$ .

\*  $p < .01$ .

Overall, the regression results largely confirm what others have found in various small-scale experiments, namely that length, syllable number, orthographic neighborhood and consistency (BOB and rime) effects tend to be larger in low- than in high-frequency words (e.g., Andrews, 1989; Chen & Vaid, 2007; Taraban & McClelland, 1987; Weekes, 1997). By and large, this pattern is correctly captured by CDP++.

### 3.7. Stress regularity

In the first experiment of Rastle and Coltheart (2000), disyllabic words were defined as being regular if they had first compared to second syllable stress. Rastle and Coltheart reported no effect of stress regularity either in error rates or RTs. Simulations with CDP++ mirrored the human data. There were no effects on errors because CDP++ made no errors. On RTs, CDP++ showed no effect of stress regularity,  $F < 1$ , a significant effect of Frequency,  $F(1, 109) = 148.31$ ,  $MSE = 8153$ ,  $p < .001$ , and no significant interaction between Stress Regularity and Frequency,  $F(1, 109) = 2.39$ ,  $MSE = 131$ ,  $p = .12$ . In terms of a quantitative comparison with the items data, the model accounted for 10.9% of the variance.

Because Rastle and Coltheart (2000) did not find a stress-regularity effect based on a first vs. second syllable stress distinction, they investigated whether there would be an effect when stress regularity was defined according to their complex algorithm (see Fig. 1). With stimuli categorized based on their algorithm, they found a significant effect of stress regularity both in errors and RTs, although this was mainly confined to low-frequency words. This latter result needs to be taken with some caution, however, as the high-frequency groups they used had only 10 items per group, and one of those high frequency items (*anode*) was removed because it produced a very high error rate and should have been classified as a low-frequency word according to CELEX.

The results showed that CDP++ produced a higher error rate for irregular than regular words (14% versus 8%). This is very similar to what Rastle and Coltheart found with irregular words, although CDP++ had a higher error rate on regular words. On inspection of the errors in the regular category, all 4 were words with second syllable stress that were assigned first syllable stress, and hence would be considered errors on irregular words using a simple definition of stress regularity. Unlike the data, CDP++ produced no effect of stress regularity on RTs,  $F < 1$ , even on low-frequency words. CDP++ also showed a very high correlation with the items data, explaining 31.3% of the variance. The results appear in Table 8.

To examine whether CDP++ could predict a stress RT effect on the low frequency items under any parameter set, the stress node naming criterion was increased to .44. This represents a strategic parameter manipulation designed to reduce the error rate of the model (see e.g., Lupker, Brown, and Colombo (1997) and Perry et al. (2007) for a discussion about strategic manipulation of response strategies). The results from this manipulation caused the low-frequency words to display a significant effect of stress irregularity,  $t(96) = 2.04$ ,  $SE = 5.57$ ,  $p < .05$ .

**Table 8**

Mean reaction times and error rates of humans (milliseconds) and CDP++ (cycles) to stress irregular and stress regular words from Experiment 3 of Rastle and Coltheart (2000).

Stress type	Rastle and Coltheart (2000)		CDP++		CDP++ (0.44 stress node naming criterion)	
	Frequency		Frequency		Frequency	
	HF	LF	HF	LF	HF	LF
<i>Reaction times</i>						
Irregular	480	543	79.9	91.5	87.8	112.0
Regular	479	515	76.7	92.4	82.5	100.6
Effect size	1	28	3.2	-0.9	5.3	11.4
<i>Error rate %</i>						
Irregular	1.71	15.46	0	14	0	0
Regular	0	1.15	0	8	0	0
Effect size	1.71	14.31	0	6	0	0

Note: HF = high frequency; LF = low frequency.

There are two other studies that investigated stress-regularity effects in words (Arciuli & Cupples, 2006; Kelly et al., 1998). Arciuli and Cupples (2006) suggested that word stress in English varies as a function of grammatical class, with nouns being irregular if they have second syllable stress and verbs being irregular if they have first syllable stress. In a single reading aloud experiment, they indeed showed a stress-regularity effect on errors (but not RTs), but the effect was much stronger for nouns (5.2% vs. 16.2% errors, for regular vs. irregular nouns) than for verbs (3.8% vs. 5.2% errors, for regular vs. irregular verbs). CDP++ showed a very similar pattern, with a stress-regularity effect on errors with irregular nouns compared to their controls (10.5% vs. 0%) but no such effect for verbs, where the model made no errors. As in the human data, CDP++ did not show a stress-regularity effect on RTs,  $F < 1$ . Thus, CDP++ was able to simulate the main pattern despite the fact that grammatical class information was not available to the network. Note, however, that there were only 20 items per cell, and since this result relies on the model making just two errors, further exploration of this phenomenon is warranted before any strong conclusions can be drawn.

Finally, we could not simulate the results from the study of Kelly et al. (1998) because their stimuli were unfortunately in Hoosier English and hence many of the items had a different stress pattern to those in CDP++'s database, which is based on Received Pronunciation, and there were also a number of proper nouns not in CDP++'s lexicon (e.g., *Corvette*). However, a superficial analysis of the remaining items showed that CDP++ made one error (4.17%) with the stress regular and five errors (25%) with the stress irregular words, thus displaying a stress-regularity effect on error rates.

### 3.8. Nonword reading and nonword stress

Using a lenient error scoring criterion, according to which a nonword response was considered correct if the phonology given by the model corresponded to any grapheme–phoneme or body–rime relationship that exists in real words, the model made 30 errors (5.1%) on Seidenberg et al.'s (1994) nonword database. A number of these errors were unlikely to be similar to those that people produce (e.g., /smu:/ for *smuice*). Some of them should therefore be considered model errors, rather than errors similar to those people produce. However, it is possible that the reading system of a person may generate something strange or phonotactically illegal that might resemble some of these atypical errors, but that their speech output system corrects or stops the output of these because they simply cannot be articulated. Thus, it is not currently possible to specify which errors should be considered model errors that are not related to those people produce and which should be considered similar to atypical responses produced but not articulated by people. For example, consider the case where the model and someone generated /fatt/, where there is a double /t/. This would have been considered an error produced by the model since people never produce it. However, whether people never generate the phonology /fatt/ is not clear. This is because, in English, /fatt/ cannot be articulated because people cannot pronounce /t/ twice without a break in the middle. Therefore, with this phonology, the speech

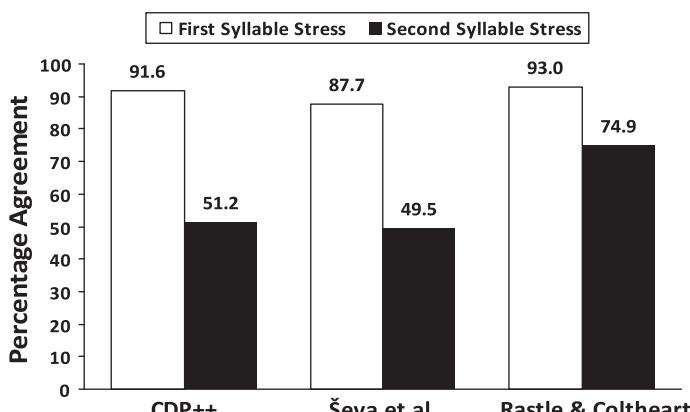
output system of the person would either have to repair the pronunciation by, for example, adding a shwa or simply not producing the second /t/ at all. If it failed to do that, it would be left with something that cannot be articulated and instead it may cause the person to stutter or produce an incomplete pronunciation which would therefore obscure the phonology that was initially generated.

To examine how CDP++ assigns stress to nonwords, the model was tested on the nonword data of Rastle and Coltheart (2000). In the study of Rastle and Coltheart, two groups were used, one where stress was predicted to fall on the first syllable and the other where stress was predicted to fall on the second syllable. Predictions were made on the basis of their complex algorithm.

When these nonwords were run through CDP++, a very similar pattern to the human data was found. CDP++ predicted that stress should fall on the first syllable 89.3% and 42.9% of the time in the two groups. Although this slightly overestimates the number of first syllable stress responses compared to people, who gave 76.7% and 23.2% of responses with first syllable stress, the results are clearly in the correct direction. In addition, the results are also closer than the predictions of Rastle and Coltheart's algorithm, which gives values of 100% and 0% for the two categories. The fits can be assessed via root-mean-square-error (RMSE) values, which show smaller errors for CDP++ than for the Rastle and Coltheart algorithm (11.68 vs. 16.44, respectively). In terms of the quality of nonword pronunciations, the model also showed reasonable generalization performance, having an error rate of only 6.3%. Nonword pronunciations were considered correct if phonemes that were given existed in other words with the same graphemes in them and were also in the linear order that would be likely to occur from their letters. Of the 54 nonwords which had a schwa in them, 2 (*ɔ:zəm, inrənt*) were assigned stress on the syllable with the schwa.

A second, and perhaps more interesting way to compare nonword stress assignment is to re-group the items based on whether first or second syllable stress was most commonly assigned to each nonword by the participants. Whether CDP++ predicts the right stress for these categorical groups can then be examined. This analysis is identical to what Ševa et al. (2009) did, and it is therefore possible to compare CDP++ with Ševa et al.'s model of stress assignment and the predictions of Rastle and Coltheart's (2000) algorithm. As can be seen in Fig. 5, CDP++ was slightly more accurate than Ševa et al.'s model even though both models used essentially the same training database. Similar to Ševa et al., CDP++ also underestimated the number of nonwords that should have had second syllable stress. On this data set, Rastle and Coltheart's algorithm had the lowest RMSE value (CDP++: 16.53; Rastle & Coltheart: 8.75; Ševa et al.: 17.32).

As Ševa et al. (2009) noted, the data set of Rastle and Coltheart (2000) may be somewhat biased by the way the items were selected. We therefore also examined the performance of CDP++ on the items in Study 2 of Kelly (2004), where the effect of the number of onset consonants on stress assignment (cf., *bedop* vs. *bledop*) was examined.



**Fig. 5.** Correct stress agreement (percentage) for CDP++, the model of Ševa et al. (2009), and the Rastle and Coltheart (2000) algorithm on the Rastle and Coltheart nonwords.

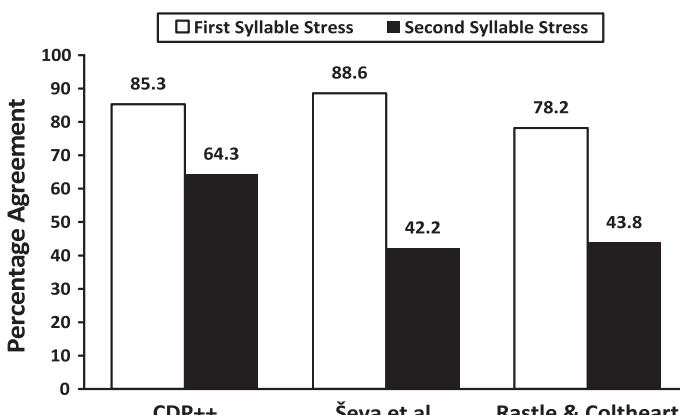
The results of CDP++ on Kelly's (2004) nonwords in terms of the percentage of first versus second syllable stress responses that were given was similar to the human data. In particular, Kelly's participants used first syllable stress 69.8% of the time and CDP++ used first syllable stress 79.8% of the time. CDP++ also predicted that words with complex onsets should be given first syllable stress more often than words with simple onsets (88.7% vs. 68.1%), which is similar to what Kelly found (78.2% vs. 62.5%). Moreover, when Kelly's nonwords were divided into two groups of items based on whether the majority of participants chose first or second syllable stress (see Fig. 6), CDP++ performed better than the model of Ševa et al. (2009) and the algorithm of Rastle and Coltheart (2000), showing the lowest RMSE value (CDP++: 12.87; Rastle & Coltheart: 20.09; Ševa: 19.63).

### 3.8.1. Modulation of vowel length

One type of pattern that is specific to disyllabic words in terms of nonword generalization is whether the first vowel in a syllable is pronounced short or long, and how consonants that occur after the vowel may affect vowel length. Waese and Jared (2006) examined this issue with three groups of nonwords that had single letter vowels in their first syllable. In one of the groups, a single consonant followed the vowel, whereas in the other two groups, two consonants followed the vowel. Of the groups with two consonants, one had consonant sequences that formed a legal onset (e.g., *gustig*) whereas the other did not (e.g., *gupdig*). The main result was that in the single consonant group, people were less likely to give short vowel responses than the other two groups (Single: 72.8%; Legal: 87.0%; Illegal: 93.8%). This is similar to what CDP++ predicts (Single: 58.2%; Legal: 81.4%; Illegal: 82.6%), although CDP++ underestimates the number of short vowels across the groups. CDP++ also made only 4.4% errors, which is a similar rate to what it made on the Rastle and Coltheart (2000) nonword set reported above. One potential reason that CDP++ may underestimate the number of short vowels is that it is only trained on monosyllabic and disyllabic words. If the model was trained on all words, it would be exposed to more short vowels, since the proportion of short compared to long vowels in multisyllabic words, at least in the first syllable of words, is more than that which the model was exposed to (62.7% vs. 53.0% according to statistics calculated from CELEX). This would be likely to cause an increase in the number of short vowels CDP++ gives given that it is sensitive to frequency.

## 4. General discussion

In the present article, we presented a full-blown model of reading aloud that deals with monosyllabic and disyllabic English words. In the spirit of the nested incremental modeling approach (e.g., Jacobs & Grainger, 1994; Perry et al., 2007), we respected the basic architecture and modeling prin-



**Fig. 6.** Correct stress agreement (percentage) for CDP++, the model of Ševa et al. (2009), and the Rastle and Coltheart (2000) algorithm on Kelly's (2004) nonwords.

ples of its precursors, CDP and CDP+ (Perry et al., 2007; Zorzi et al., 1998a). As dictated by the nested incremental modeling approach, the new model was tested against the old benchmark effects that motivated the development of the previous models as well as novel benchmark effects specific to disyllabic reading aloud, including the effects of syllable number, stress regularity, and nonlexical stress assignment. Before going into a detailed discussion of the novel effects, the main results can be summarized as follows:

1. CDP++ has been successfully scaled up, and, using its standard parameter set, it reads aloud more than 32,000 words in its lexicon with a mispronunciation rate of less than one percent. While performance of some connectionist models might deteriorate when being up-scaled to a real-size corpus (e.g., Feldman-Stewart & Mewhort, 1994), CDP++ accounts for as much variance as its monosyllabic precursor (CDP+) on the four critical large-scale databases of monosyllabic words (i.e., Balota & Spieler, 1998; Spieler & Balota, 1997; Treiman et al., 1995; Seidenberg & Waters, 1989).
2. CDP++ is fully backwards compatible with its predecessors as it is still able to simulate all major monosyllabic benchmark effects that motivated the development of the earlier models. The only slight difference was that the model failed to show a significant body neighborhood effect, unlike CDP+.
3. CDP++ accounts for more item-specific variance in naming latencies on the ELP database of Balota et al. (2007) than any other model has, explaining over 49% of the variance on a restricted selection of monomorphemic monosyllabic and disyllabic words (Yap & Balota, 2009) when onset coding is included. Given that the reproducible variance in a large-scale database is probably no more than 40% (see Rey, Courrieu, Schmidt-Weigand, & Jacobs, 2009) the present result can be taken as a major achievement.
4. CDP++ is sensitive to many of the same variables that affect human latencies, that is, when variables such as frequency, syllable number, letter length, neighborhood, and consistency are regressed onto the model's latencies, the obtained correlations are strikingly similar to those of humans (see also Yap & Balota, 2009). This is a new model test that is more fine-grained than the simple percentage of variance accounted for. CDP++ passed this test rather well and it also successfully simulated interactions between all of these variables and frequency.

**Table 9**  
Disyllabic benchmark effects.

Name of effect	Benchmark data set	Description
Large-scale databases	Balota et al. (2007), Yap and Balota (2009), and Chateau and Jared (2003)	Models should account for a large portion of item-specific variance on large-scale databases
The Yap and Balota test	Balota et al. (2007), Yap and Balota (2009), and Chateau and Jared (2003)	$\beta$ -coefficients for important lexical variables and theoretically meaningful interactions should be similar in the human and model data
Syllable number	Balota et al. (2007) and Yap and Balota (2009)	Disyllabic words are slower to read aloud than monosyllabic words, even when matched on other characteristics
Syllable number by frequency interaction	Balota et al. (2007) and Yap and Balota (2009)	The effect of syllable number is larger for low-frequency than for high-frequency words
Consistency/regularity	Jared and Seidenberg (1990)	Inconsistent or irregular words take longer to read aloud than regular/consistent controls
Stress regularity	Yap and Balota (2009) and Chateau and Jared (2003)	Words with second syllable stress yield a higher error rate than words with first syllable stress
Stress by onset complexity interaction	Kelly (2004)	Nonwords with complex onsets are more likely to be given first syllable stress than nonwords with simple onsets
Modulation of vowel length	Waese and Jared (2006)	Nonwords with single letter vowels in their first syllable are less likely to be given short vowel answers if only a single consonant follows the vowel

5. CDP++ is a full-blown model of reading aloud that incorporates a mechanism for stress assignment in its regular processing dynamics. This makes it possible to investigate how stress information interacts with lexical and nonlexical processing in reading aloud.

#### *4.1. Novel benchmark effects for disyllabic reading aloud*

There are far fewer studies on the reading aloud of disyllabic words than there are on monosyllabic words. A list of these benchmark effects can be found in [Table 9](#) (a file with items and item data can be downloaded at <http://ccnl.psy.unipd.it/CDP.html>). CDP++ was confronted with all of these different item sets and the results were globally satisfactory. In particular, CDP++ was able to capture the effect of syllable number, even when all other likely confounding variables were controlled for. This is an important finding given that the model does not contain a mental syllabary. This finding clearly counters the claim that “any model of lexical access has to incorporate a syllable level of representation or include the syllable as a sublexical unit in processing” ([Álvarez et al., 2001, p. 553](#)). However, it should be noted that whilst CDP++ may have no explicit syllable representations, the way that both the training and running of the model occur means that information is available that allows both graphemic and phonemic representations to be aligned based on where syllable breaks either do occur (training mode) or are likely to occur (running mode). Thus, knowledge about syllable breaks which guides the placement of graphemes is used by the model, although this does not amount to explicit syllabic representations that are retrieved at some stage of processing by the model.

#### *4.2. Consistency and regularity*

CDP++ was also able to account for consistency and regularity effects both in small-scale experiments as well as large-scale studies. Note, however, that a lack of item significance (which is critical for computational models) affects several studies of consistency examining multisyllabic words (with the notable exception of those based on the ELP database). This may reflect the more general issue of how to compute consistency measures when more than one syllable is involved (e.g., [Jared & Seidenberg, 1990](#); [Taft, 2001](#)). In particular, as can be seen in both the database analyses of [Chateau and Jared \(2003\)](#) and [Yap and Balota \(2009\)](#), consistency effects can be found at many different levels and trying to investigate effects of consistency at one level whilst trying to balance stimuli on all other level in small-scale experiments may be very difficult.

An important result that differentiates CDP++ from DRC is that it displayed a consistency effect on the second vowel in disyllabic words, whereas DRC predicts that sublexical phonology only affects the first few phonemes ([Rastle & Coltheart, 1999](#)). This was not simply due to some unknown confound in the items it was tested on (i.e., Experiment 4 of [Chateau & Jared, 2003](#)), because if the model is run without sublexical phonology, then no significant difference between groups is found (Irregular vs. Regular: 97.5 vs. 99.5 cycles,  $t < 1$ ). This difference therefore reflects differences in the parameterization of the models and the efficiency and functioning of the sublexical route. In this respect, the speed at which CDP++ processes letters is much faster than DRC (10 cycles vs. 17) and the response threshold is such that it also takes a longer amount of time for words to be named. As a consequence, the number of letters that the sublexical route processes before the word is read aloud is larger in CDP++ than DRC and therefore a greater amount of sublexical phonology is activated. Such a difference in parameterization is also part of the reason that CDP+ displays length effects on words, also unlike DRC.

One aspect of the results where the model differed was that it generally under-predicted the strength of the BOB effect. The model therefore behaves more like [Taft's \(2001\)](#) average readers, who show stronger syllable than BOB effects, compared to his good readers, who display the opposite pattern. However, there is no principled reason for why some version of CDP++ could not simulate BOB effects even when using syllabically aligned graphemes. This is because graphemes do not necessarily have to be contiguous for CDP++ to learn relationships between them. Thus, if there are cases where the disambiguation of a grapheme (generally the vowel) can benefit from graphemes in a different syllable, then it may be possible for the model to use those graphemes in different syllables to help in this process. If, for example, the consonant graphemes of a word that would form a BOB division occur as the onset of another syllable, they can still help in differentiating the vowel. Indeed, as long graphemes

consistently co-occur in the same place and consistently map to the same phonology, CDP++ should be able to learn relationships between those graphemes and phonology, no matter what position in the template they have been given.

#### 4.3. Stress

One of the main extensions that allowed CDP++ to capture new disyllabic data was the addition of sublexical stress and stress output nodes. Sublexical stress nodes were added to the sublexical network to allow it to predict which of the two syllables should be stressed when presented with a disyllabic stimulus. During training, these nodes were activated on the basis of lexical phonology, and the relationship between the graphemes and the sublexical stress nodes was learnt in exactly the same way as the relationship between graphemes and phonemes. Later, in running mode, the activation of these nodes represents a prediction made by the sublexical network about which syllable should be stressed, irrespective of the lexical status of the input (i.e., word or nonword). The final stress assignment, however, is determined by the activation of the two stress output nodes that were added to the output level of the model. These nodes receive input both from the sublexical stress nodes and phonological lexical nodes where the stress pattern of each word is stored. Competition between these two sources of information was modeled using the standard interactive-activation equations. Apart from determining which syllable should be given stress, the stress output nodes can also affect when the model finishes running, with processing only being terminated once the amount of activation in at least one of the nodes has risen above a predetermined threshold.

With these straightforward extensions, CDP++ was able to simulate a number of quite complex stress effects in reading aloud. These included error effects found mainly with words with second syllable stress on the large databases, error effects on a number of small-scale experiments, and the non-word stress data of [Kelly \(2004\)](#) and [Rastle and Coltheart \(2000\)](#). These simulations were all done with a default stress node naming criterion of 0.1. This value was chosen so that stress typically does not affect RTs. This choice was made because most small-scale experiments showed no reliable effects of stress on RTs. In addition, in large-scale databases, stress regularity has a reliable effect on errors but not RTs ([Chateau & Jared, 2003](#); [Yap & Balota, 2009](#)). It thus appears reasonable to hypothesize that stress regularity does not typically affect response latencies.

To simulate stress effects on RTs, the stress node naming criterion had to be increased, which means that additional time is given to the system before read-out occurs. A strategic change of the naming criterion to 0.44 allowed CDP++ to simulate the RT effect of [Rastle and Coltheart's \(2000\) Experiment 3](#), which was the only experiment reviewed that showed such an effect. Such a strategic adaptation is based on the idea that the response criterion may be increased when especially hard items are used. This was the case with Rastle and Coltheart's items, where both the model and human data had a very high error rate (over 15% in the low frequency irregular category in humans and 14% with the model).

An interesting aspect of CDP++ is that it often predicted the correct stress patterns despite not having any explicit information about the particular stress regularity metric being used (e.g., grammatical class, morphological principles). This highlights the importance of understanding how a number of highly intercorrelated variables determine how stress is assigned and how this may cause some variables to appear to have a causal effect even when they do not (see [Arciuli & Cupples, 2006](#)). For example, effects that seem to be caused by one particular variable, such as grammatical class, might be due to other highly intercorrelated variables. Having an implemented model without many linguistic assumptions built into it can certainly help in this area of research by providing a null model against which more linguistically-motivated stress effects can be tested.

The results also suggest that CDP++ offers a far simpler and hence parsimonious explanation for stress assignment than that of [Rastle and Coltheart \(2000\)](#), who needed to include morphology and a number of complex decision processes into their algorithm. This is theoretically very important as both [Arciuli and Cupples \(2006\)](#) and even Rastle and Coltheart have noted that using morphology as a source of information in the sublexical stress assignment process is potentially very problematic.

One discrepancy between the stress simulations of CDP++ and the human data was that CDP++ assigned first syllable stress to nonwords more commonly than did people. The modeling work by [Ševa](#)

et al. (2009) showed exactly the same problem. We do not think that this divergence from the human data reveals a serious flaw with the model, however, because words with first syllable stress are overrepresented in disyllabic words (Ševa et al., 2009). If CDP++ were trained on a full multisyllabic database, it is likely that this bias would disappear. A simple way to improve the results could also be to weight the evidence from the two stress nodes differently. If, for example, the weighting of the second stress node is increased compared to the first, it is possible to increase the performance of the model on Experiment 2 of Rastle and Coltheart (2000) from 51.2% to 73.5% on nonwords typically given second syllable stress, with a much smaller decrease in performance on nonwords typically given first syllable stress (from 91.3% to 82.3%).

An interesting aspect of the simulations was that CDP++ did better than the model of Ševa et al. (2009), even though both use a similar assumption that a lot of information about stress can be learnt from simple spelling–sound correspondences. The most likely reason that CDP++ performs better is that CDP++ uses graphemes as sublexical input, whereas the model of Ševa et al. (2009) uses letters. Graphemes are likely to have two advantages over letters. One is that they allow better generalization since the relationships between graphemes and phonemes are less dispersed than those between letters and phonemes. This is one of the reasons why nonword reading was so much better in CDP+ than in CDP (Perry et al., 2007). A second reason is that English syllables are “weight-sensitive” (e.g., Gordon, 2006). This means that one factor influencing whether a syllable is likely to be assigned stress is the number of phonemes in its coda. With CDP++, this can be learnt simply from the position graphemes occur in syllables because, if a consonant grapheme occurs in a coda position after other consonant graphemes (i.e., latter coda positions), the syllable is very likely to have more than one coda phoneme and hence is more likely to attract stress than a syllable that does not. Alternatively, when letters are used, this information is much less reliable because there are many sequences of letters that map onto only one phoneme. This means that even if a consonant occurs after others in the coda, it does not always mean that the syllable is likely to have many phonemes. For example, if graphemes are used as input, then when a ‘k’ occurs as the second grapheme of a coda, it generally maps to the second phoneme of a word (e.g., *silk, frisk, mink*). Its position therefore gives some idea about the likely number of coda phonemes in a word (i.e., at least 2). Alternatively, if letters are used as input, then this information is much less reliable, because the ‘k’ not only commonly maps to words with two phonemes, but words with one also (e.g., *pick, flock*).

Despite the overall success of the stress simulations, it is important to note that many cognitive components related to stress at a prosodic level are not integrated into the model (see e.g., Hayes (1995) for a cross-language discussion of metrical systems). Such components may well affect the way stress is assigned in some circumstances via feedback, such as when stress patterns become predictable in the context of sentences (e.g., Arciuli & Cupples, 2003) or word lists (e.g., Roelofs & Meyer, 1998). The extent to which such feedback from other cognitive systems affects stress assignment, and hence knowing what the balance between orthography and phonology in the assignment of stress is, remains to be determined. However, it is clear that words can be assigned stress phonologically, even in the silent reading process (Ashby & Clifton, 2005), and thus the interaction between orthography and phonology is likely to be important.

Finally, the stress mechanisms of CDP++ are also compatible with neuropsychological data from patients with acquired dyslexia that make stress errors. For example, Miceli and Caramazza (1993) reported the case of an Italian speaker, CLB, who read words aloud with nonword-like stress assignment, thereby producing many suprasegmentally incorrect responses to irregularly stressed words (i.e., a stress-regularity effect). In contrast, CLB was able to read nonwords and assign stress to them in a way similar to that of normal participants. Miceli and Caramazza interpreted this pattern of results as evidence that CLB was able to generate syllabic phonology via the use of a sublexical mechanism, and then generate the correct stress pattern from the syllabic structure of the phonology. CDP++ offers an additional possibility to this, which is that CLB’s stress assignment was based simply on orthographic patterns, rather than syllabic structure. Thus, rather than assuming that stress is always generated based on syllabified phonology, stress may be generated simply by examining the stress nodes that are activated from orthographic information (see also Arciuli & Cupples, 2006; Kelly et al., 1998).

#### 4.4. Syllable number

One of the important effects that CDP++ captures is the effect of syllable number. As can be seen from the results section, CDP++ closely matches the human data. There are two different ways that this effect arises with CDP++, one is due to the graphemic parsing processes whereas the second is due to vowel consistency.

In terms of the parsing process, words with two syllables are more difficult to process than words with one because when the parser has only processed a single vowel in disyllabic words, it still places graphemes in the first syllable of a word, which is generally incorrect. It is not until the parser processes the second vowel that these graphemes can be correctly placed. Such behavior slows the processing of graphemes in the second syllable down due to this delay.

The second factor is that the vowel of the second syllable is highly inconsistent (Chateau & Jared, 2003). This implies that words that have two vowels are more likely to be slower to process than words that have one because both have the potential of slowing the naming process down. This effect can still occur with CDP++ even though the parsing process works from left to right because the speed at which letters are processed in CDP++ is fast enough to allow sublexical phonology to be processed in the second syllable, although the extent that this happens is dependent on the speed that lexical phonology arrives, and hence an interaction with frequency is found.

#### 4.5. CDP++ versus other models

As discussed in the introduction, CDP++ can be compared to two implemented models of multisyllabic word reading: the MTMM of Ans et al. (1998) and the Junction model of Kello (2006). Unfortunately, the comparison with the MTMM is rather limited because the model is only implemented in French. Although Ans et al. provided simulations of some English benchmark effects using French stimuli, it is questionable whether this strategy is appropriate. For example, Ans et al. showed that the MTMM produced a frequency by regularity interaction that was meant to mirror the English data. However, it is unclear whether the model should show such an interaction in French because previous empirical work showed no interaction between frequency and regularity in French (Content, 1991; Ziegler et al., 2003). Thus, the model is simulating “English” effects with French words, without ever questioning whether the French data are really the same as the English data for a given set of words. Leaving this issue aside, at least for current model comparisons, the main problem is that the performance of the MTMM cannot be assessed on large-scale English databases and English nonword reading performance, both of which present key benchmarks for model comparisons.

Another difficulty with evaluating the MTMM is that it has no mechanism to deal with stress. While this might not be a problem for French, stress is certainly relevant for English. This is especially problematic with nonwords, as the MTMM assumes that some nonwords may be read aloud in “analytical mode”, in which single syllables are presented to the model one at a time. It is therefore not clear how the model could generalize stress knowledge learnt from words, which are not learnt one syllable at a time, to nonwords read aloud in analytical mode.

A fundamental difference CDP++ has compared to the MMTM (and indeed to most other models) is that with the MMTM, the same sequence of letters are always processed in the same positions. This causes an obvious problem for stimuli such as heterophonic homographs (e.g., *crooked* vs. *crooked*), for which the same orthographic pattern maps onto two phonologies (/krukt/ and /kru.kid/). These patterns are comparatively difficult to learn for the MTMM because the learning of one pattern will interfere with the learning of the other. In CDP++, there is no such interference because identical letter strings are not necessarily processed in the same positions. In *crooked*, the -ked is put in the first syllable and maps to /kt/, while in *crooked* it is put in the second syllable and maps to /kid/. Thus, the two models make different predictions as to how these types of stimuli are processed. The MTMM predicts a massive interference effect, while CDP++ does not. Similar problems would also occur at the subword level when identical letter sequences are processed in different words, such as the -gged in *jagged* and *bagged*. This is because the letter -e can be treated by CDP++ like a coda consonant in the first syllable (e.g., *bagged*; see also Plaut et al. (1996), who also allow -e to be treated as a coda consonant) or like a vowel in the second syllable (e.g., *jagged*), causing two different orthographic

segmentations. Alternatively, in the MTMM, the letter –e would be coded in the same fixed order and place, and hence there would be interference in learning from the same letters having to activate two different phonologies.

The Junction model can be compared to CDP++ both qualitatively and quantitatively. In terms of quantitative performance on large-scale databases, the Junction model does a good job, accounting for up to 30% of the variance on the monomorphemic words of the ELP. It is also sensitive to the same variables that affect human latencies in these databases (Yap & Balota, 2009). However, the Junction model has problems with nonword reading, performing far below the level of skilled readers, and therefore any results should be seen as tentative until a final version that fixes this problem is released.

One fundamental difference between CDP++ and other models is the way serial effects are produced. In CDP++, serial effects occur in part because of a processing assumption within the model (left-to-right grapheme parsing), whereas others (e.g., Kello, 2006; Plaut et al., 1996; Seidenberg & McClelland, 1989) assume that they are caused by peripheral effects related to either speech production or visual encoding. Whilst length effects being caused by peripheral factors could potentially affect CDP++, there are a number of strong reasons against believing that serial (or serial-like) effects are only due to peripheral processes that occur in either speech production or early visual processing. First, serial effects disappear in delayed naming (Weekes, 1997). If they were due to peripheral processes of production, they should persist in delayed naming. Second, the size of the length effect found in speech production tasks (e.g., Roelofs & Meyer, 1998) is relatively small compared to the size of the length effect found in some reading aloud tasks (e.g., Weekes, 1997). Third, length effects are absent or much smaller in lexical decision compared to naming tasks (e.g., Balota et al., 2004). If length effects were caused by visual encoding processes, length effects should be seen in word recognition tasks other than just reading aloud. Fourth, the size of length effect is different in German and English even when almost identical items are used in the two languages (Ziegler et al., 2001). This suggests that serial effects occur in the mapping between orthography and phonology rather than in visual input or phonological output processes, which would have been the same in the two languages. Finally, serial effects are stronger for nonwords than for words even if visual input and phonological output is held constant, the first of which is possible in Serbian and the second of which is possible in Japanese (Rastle, Havelka, Wydell, Coltheart, & Besner, 2009).

#### 4.6. Criticisms and caveats

One potential criticism to the present approach is that it is more concerned with predicting variance and a large number of effects than understanding broader principles in word recognition and reading aloud. A related criticism is that the superior performance of our model compared to other models is simply due to the fitting of a large number of free parameters (Sibley et al., 2010). We think that these criticisms are neither correct nor justified.

First, the present model is based on the principles of the connectionist dual route approach to modeling reading aloud (Zorzi, 2010), which argue that two distinct processes are needed, a sublexical process that implements a linear mapping between orthographic and phonological patterns and a lexical process that retrieves word-specific information possibly in a non-linear and mediated fashion (Zorzi et al., 1998a). As such, CDP++ stands in the long tradition of dual-route theories of reading. Our work further shows that the same principles can be used to read words with more than one syllable and to handle stress. Finally, we demonstrated that a linguistically motivated principle, onset maximization, in combination with internal network dynamics are sufficient to solve the tricky problem of segmenting disyllabic letter strings.

Second, here and in our previous work (Perry et al., 2007), we have clearly defended the position that a few theoretically important effects (e.g., consistency, length) can falsify a model regardless of how many other effects or how much variance this model can account for. Fortunately, however, it turns out that the model that currently simulates most of the theoretically relevant effects – CDP++ – is also the one that accounts for the largest proportion of the variance in large-scale databases.

Third, a closer inspection actually shows that there are hardly any parameters that were fit in CDP++. Indeed, the lexical route is almost identical to the interactive activation model as implemented

by Coltheart et al. (2001) and the sublexical route contains only a few parameters, and these are “set” rather than “fit”. Indeed, the parameters are set in a way such that there is a balance between lexical and sublexical processing. No parameter optimization algorithms were used in finding the best parameter set and all parameter changes are fully interpretable. Most importantly, there is no parameter fitting of individual experiments, data sets, or large-scale studies, even if this could be easily justified (e.g., list composition, item difficulty, reading experience, etc.). Thus, parameter fitting is minimal and highly constrained in the nested modeling approach. Note also that there are many effects the model captures where there are essentially *no* parameters that can be modified to change the model behavior (e.g., nonword generalization performance).

Finally, some parameter-heavy processes are simply “inherited” from precursor models but could actually be replaced by simpler procedures. For example, a version of CDP+ where the lexical route was reduced to the provision of a frequency-weighted, feedforward activation of lexical phonology showed almost the same performance as the full-blown model, both qualitatively and quantitatively (Zorzi, 2010; see also Perry et al., 2007). Therefore, the large number of parameters that are needed for the interactive activation part of the model could be reduced substantially even in CDP++. We have not done this because the original lexical route is still required to simulate some effects, such as pseudohomophone effects (e.g., McCann & Besner, 1987), that depend on feedback or interactions between the various processing layers of the interactive activation model. Moreover, the current lexical route implements a mechanism for visual word recognition that has been widely used to account for perceptual identification and lexical decision data (Coltheart et al., 2001; Grainger & Jacobs, 1996; McClelland & Rumelhart, 1981).

#### 4.7. Limitations and future directions

An apparent limitation of the present work is its focus on disyllabic rather than all multisyllabic words. This was a deliberate choice in order to isolate and fully understand the problems of reading aloud disyllabic words before moving onto more complex systems. However, there is no principled reason why one could not simply keep on extending the current coding scheme to more than two syllables. That is, instead of using a two-syllable template, a multisyllabic template could be used. This is the strategy used by the MTMM, which allows up to 5 syllables to be processed at the same time. A similar scheme could be used with CDP++, where the grapheme parser could simply work across syllables in the way that it works across the current two syllables – that is, when a new vowel is identified, it could try to work out the best slots for graphemes to go in the new syllable.

One might argue that duplicating the syllabic template for each additional syllable is not very elegant as one may end up with a “monster-template”. However, we believe that this is simply an implementational detail, one that allows us to efficiently code and align letter strings. The real issue we believe is important in extending the model even further is that one has to propose a mechanism for integrating information across saccades. In particular, given that the mean saccade size in silent reading is around 8 letters (Rayner, 1998), it means that most words with more than two syllables are likely to be fixated more than once. Integrating information across saccades and determining the effect it has on processing is a difficult problem for all current models, although some proposals have been made on how that might be achieved (Plaut, 1999).

A second issue to do with integrating information is how the broader language system interacts with the reading system (see e.g., Arciuli & Cupples, 2003; Ashby & Clifton, 2005). This is important in terms of stress assignment because multisyllabic words are generally believed to be organized into stress or prosodic feet (e.g., Burzio, 1994; Hammond, 1999; Hayes, 1995; Selkirk, 1980), which specify how stress is assigned to small groups of syllables within words. How the reading system can influence the assignment of such feet, or indeed whether the sublexical system actually produces the phonology of feet rather than whole words, is currently not well investigated.

A third apparent limitation of the present work is our focus on simulating reading aloud rather than including other tasks, such as lexical decision or perceptual identification. Computationally speaking, reading aloud is a harder problem than lexical decision because the model must produce the exact phonological output for words and nonwords, whereas lexical decisions can be based on partial or underspecified information. Nevertheless, it is important to note that CDP++ can make lexical

decisions to the same extent as DRC (Coltheart et al., 2001) or MROM (Grainger & Jacobs, 1996) can make lexical decisions. That is, “yes” decisions in these models are based on a read-out that takes into account word specific and global activation in the orthographic lexicon, whereas “no” decisions are based on a variable temporal threshold. Unfortunately, the temporal threshold mechanism for “no” responses has been heavily criticized. In particular, Wagenmakers, Ratcliff, Gomez, and McKoon (2008) argued that the response mechanism should be modeled as a diffusion process rather than a deadline mechanism. Thus, until this debate has been settled, we decided to refrain from implementing a decision mechanism in the present version of the model.

One area where CDP++ (and indeed all other major models of reading aloud) remains underdeveloped is in simulating errors and individual differences. At present, CDP++ is designed to explain the “average” pattern of findings, rather than individual data. This has a number of consequences. One of the most important is that the parameter set is deliberately chosen so that almost all words are read correctly. This is true both of the phonology of words generated and the stress they are assigned. However, this also means that the model is not good at simulating errors. Of course, one could change the parameters to make the model make more errors. If this were done, however, then the model would no longer simulate “average” results. The reason for this is that if an item is read incorrectly by CDP++, it is clearly not behaving like the *average* response to that item (which is not an error), and this is especially true for most small-scale experiments where items with high errors rates are usually excluded. One solution to this problem would be to conduct individual simulations, in which parameters are allowed to vary from one subject to the other, thus occasionally creating an error for one subject but not for another. This approach is particularly interesting when applied to modeling errors in atypical readers (e.g., developmental dyslexics, see Ziegler et al., 2008). A second approach could be to train models under different conditions and then examine their effect on performance (e.g., Zevin & Seidenberg, 2006).

An important direction for future development of CDP++ is that of increasing the role of learning, which is currently limited to the sublexical spelling-to-sound and spelling-to-stress mappings. First, learning in CDP++ could be augmented to include the identification of graphemes from the letter level and their assignment to the graphosyllabic template. Second, lexical representations could also be learnt as part of the training phase rather than set by hand (see Zorzi (2010) for discussion). The Interactive Activation network that forms the lexical route was perfectly suitable for our purposes, because all it needs to do for the model to investigate effects related to sublexical processing is to produce a simple frequency-weighted activation (Perry et al., 2007; Zorzi, 2010). However, there is no reason to think that better results could not be obtained with a different scheme that includes the learning of word forms and uses a letter level based on more modern data (see e.g. Davis and Bowers (2006) for a review of just the letter level alone), and such changes remain to be investigated within a full-blown model.

## 5. Conclusions

In the present paper, we have successfully extended our modeling work to disyllabic words, and have successfully dealt with problems such as syllabification and stress assignment. An executable version of the model is available at <http://ccnl.psy.unipd.it/CDP.html>. This model can be used not only to test novel predictions about reading aloud, such as effects of stress typicality, syllable frequency, and syllable neighborhood, but can also be used as a null model for the investigation of processes that are not implemented in the model, such as the effects of morphology, semantics, or emotional valence. Most importantly, CDP++, with its lexicon of over 32,000 words, is a notable example of the successful scaling-up of a connectionist model to a size that more realistically approximates the human lexical system.

## Acknowledgments

This research was supported by a Swinburne Staff Development grant to CP and by grants from the Australian Research Council (DP0985815) to CP and the European Research Council (210922-GENMOD)

to M.Z. J.Z. was supported by an *Alexander-von-Humboldt Fellowship*. We thank Debra Jared, Melvin Yap, Padraic Monaghan and Nada Ševa for providing various data sets and measures and a number of reviewers for useful comments. Correspondence concerning this article can be sent to ConradPerry@gmail.com. CDP++ is available for download at <http://ccnl.psy.unipd.it/CDP.html>.

## **Appendix A. Graphemes used**

Multi-letter graphemes used in the graphemic buffer:

*Onsets:* ch gh gn ph qu sh th wh wr kn

*Vowels:* air ai ar au aw ay ear eau eer ea ee ei er eu ew ey ier ieu iew ie ir oar oor our oa oe oi oo ou or ow oy uar ua ue ui ur uy ye yr

*Codas:* tch ch ck dd dg ff gh gn ll ng ph sh ss th tt zz nn gg pp bb ff mm cc rr mb

## **Appendix B. Parameters used in the model**

Lexical route

*Features*

Feature to letter excitation: 0.005

Feature to letter inhibition: -0.8

*Letters*

Letter to letter inhibition: -0.3

Letter to orthography excitation: 0.018

Letter to orthography inhibition: -0.8

*Orthographic lexicon*

Orthography to orthography inhibition: -.1

Orthography to letter inhibition: 0

Orthography to phonology excitation: 1.9

Orthography to letter excitation: 0

*Phonological lexicon*

Phonology to phonology inhibition: -.12

Phonology to phoneme excitation: 0.09

Phonology to phoneme inhibition: -0.125

Phonology to orthography excitation: 1.5

*Phoneme output buffer*

Phoneme to phoneme inhibition: -0.005

Phoneme to phonology excitation: 0.09

Phoneme to phonology inhibition: -.18

*Parameters used in the sublexical route*

Grapheme parsing cycles per letter: 10

Sublexical network to phoneme output buffer/stress output node activation: 0.072

Level of activation which a letter must be over before graphemic parsing begins: .30

Temperature ( $\tau$ ) in the sublexical network: 3

Learning rate ( $\varepsilon$ ) in the sublexical network: 0.05

Dead node level: 7.5

*Word stress parameters*

Stress node naming criterion: 0.1 (unless otherwise stated)

Phonological lexicon to stress output node excitation: .037

Phonological lexicon to stress output node inhibition: -.023  
 Stress output node to stress output node lateral inhibition: -.11

*Overall parameters*

Overall activation rate: 0.3

Lexicon frequency scaling:  $0.2 \times (\log(\text{word frequency} + 2)/\log(\text{maximum word frequency} + 2))$

Phoneme naming activation criterion: 0.69

Cycle-to-cycle stopping criterion: 0.0024

## Appendix C. Orthographic information and the letter –e

A simple two-layer network with identical dynamics to the CDP++ sublexical network was used to determine how the letter –e is processed (i.e., whether it is treated as a coda consonant or whether it is treated as the vowel of the second syllable). The network was trained on all words in the database that have the letter –e as a second vowel grapheme (7946 words). Training occurred between the letter level (the input representation) and two output nodes. The two output nodes represented whether the letter –e should go in the first or second syllable. The letter information was organized into wickelgraphs (letters triplets; see e.g., Seidenberg & McClelland, 1989). Wickelgraphs were used simply as a convenient way to represent letters and relationships between them. Only the information about letters that occur between the first and the second vowel (including the latter) were used to form the wickelgraphs.

The network was trained for 30 cycles. After training, the predictions the network made about whether the letter –e should go in the first or second syllable were examined. This was done with all words that the network was trained on and was done by assigning the letter –e to the syllable corresponding to the output node in the network that had the stronger activation. Based on this simple criterion, the model was able to correctly classify all but 18 words. Since some letter patterns are ambiguous, in that they can lead to either monosyllabic or disyllabic pronunciations, a correct classification was determined by examining whether an extant coda from a monosyllabic database existed that completely overlapped the coda letters of the word being examined. If it did, the answer was considered correct even if the word was disyllabic, because the pronunciation could potentially be monosyllabic if it were a nonword. For example, the word *naked* is disyllabic, but the coda is used in monosyllables, e.g., *baked*. Thus, even though the network predicts this word should be monosyllabic but *naked* is disyllabic, this was not considered an error in evaluating whether the letter –e can be assigned to a syllable position where it commonly goes. Note that for all of the simulations reported elsewhere, if the parser assigned a letter to the incorrect syllable, and this was not corrected by lexical phonology and hence the number of syllables output by the model was wrong, this was still considered an error.

It is worthwhile noting here that similar results can be achieved using only a memory of the graphemes the parser has encountered and an attentional window that has a greater number of letters than the largest grapheme. The larger attentional window is needed since some identical letter sequences can occur within monosyllabic and disyllabic words and whether they should force the creation of one or two syllables can only be determined by either the word ending or letters to the right of the sequence. For example, the –es sequence often occurs in monosyllable (e.g., *fines*), but the –est sequence typically occurs in disyllables (e.g., *finest*). Thus, the only way the –e can be correctly classified in this case is to use a context that includes the two letters to the right.

## Appendix D. Monosyllabic effects and dyslexia

### D.1. Frequency, lexicality, length by lexicality

Weekes (1997) conducted a classic experiment examining length, lexicality, and frequency. He found effects of all of these, and, importantly, an interaction between length and lexicality. Our first analysis using his items with CDP++ was done using Length and Lexicality as factors in an ANOVA,

and log Frequency as a covariate (nonwords were counted as having a frequency of zero). The results showed that CDP++ showed main effects of Length,  $F(3, 280) = 69.88, MSE = 5373, p < .001$ ; Lexicality,  $F(1, 280) = 1151.391, MSE = 88,533, p < .001$ , as well as an interaction,  $F(3, 280) = 7.97, MSE = 613, p < .001$ . We also compared just the high-frequency words to the low-frequency words, and CDP++ showed an effect of this also,  $t(195) = 8.96, SE = .98, p < .001$ . The model also showed a reasonable correlation with the items data, explaining 7.3% of the word and 20.3% of the nonword latency variance. (CDP++ cycles, High-Frequency Words (3–6 letters): 62.4, 68.7, 72.7, 77.4; Low-Frequency Words (3–6 letters): 71.3, 77.2, 81.6, 85.7; Nonwords (3–6 letters): 99.7, 106.0, 116.3, 127.6).

#### D.2. Frequency $\times$ regularity, word consistency

A benchmark experiment for regularity chosen by Coltheart et al. (2001) comes from the study by Paap and Noel (1991). Whilst this study was run under dual-task conditions, it has been replicated many times even under conditions of very light memory load (e.g., Bernstein & Carr, 1996). CDP++ showed the same result, displaying a significant interaction,  $F(1, 75) = 5.85, MSE = 245, p < .05$  (CDP++ cycles, High Frequency Irregular: 66.0; High Frequency Control: 64.2; Low Frequency Irregular: 85.9; Low Frequency Control: 77.0). A *t*-test examining the high frequency irregular words versus their controls was not significant,  $t < 1$ , unlike a similar comparison with the low-frequency words,  $t(37) = 4.03, SE = 2.20, p < .001$ .

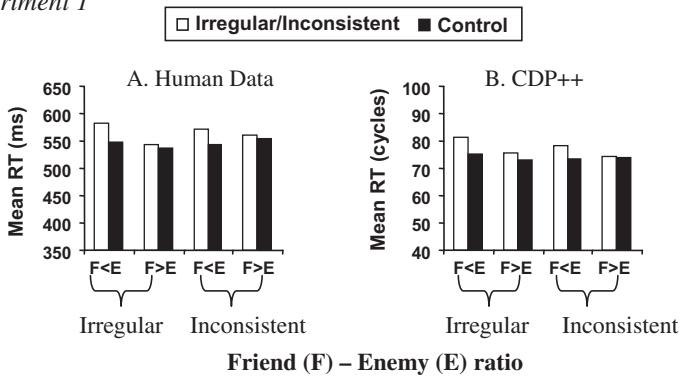
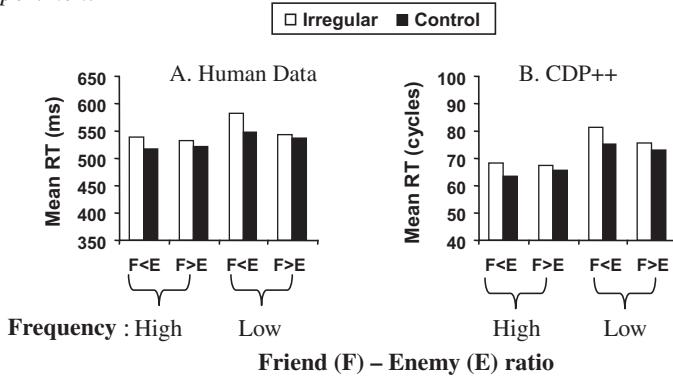
We also tested CDP+ on other data sets, in particular the carefully controlled study of Jared (2002). As can be seen in Fig. D1, where the results of Jared's (2002) Experiment 1 and Experiment 2 are shown along with those of CDP++, the model displayed essentially the same effects as the data. In the crucial first experiment, the model, like the human data, showed an effect of regularity and consistency with low-frequency words, but only with words that had a greater token count of enemies ( $E$ ) than friends ( $F$ ) (Irregular ( $E > F$ ),  $t(38) = 3.00, SE = 2.19, p < .01$ ; Irregular ( $F > E$ ),  $t(37) = 1.48, SE = 1.90, p = .15$ ; Inconsistent ( $E > F$ ),  $t(38) = 3.18, SE = 1.59, p < .005$ ; Inconsistent ( $F > E$ ),  $t < 1$ ). In the crucial second experiment, like the human data, CDP++ showed a regularity effect with both high and low-frequency words that was restricted to words with more enemies than friends (High Frequency ( $E > F$ ),  $t(37) = 2.22, SE = 1.98, p < .05$ ; High Frequency ( $F > E$ ),  $t < 1$ . Low Frequency: Irregular ( $E > F$ ),  $t(38) = 3.00, SE = 2.19, p < .01$ ; Irregular ( $F > E$ ),  $t(37) = 1.48, SE = 1.90, p = .15$ ). The quantitative performance of CDP++ was also excellent across the four experiments, accounting for 29.0%, 32.4%, 46.4%, and 40.0% of the variance. This is very similar to CDP+.

#### D.3. Nonword consistency

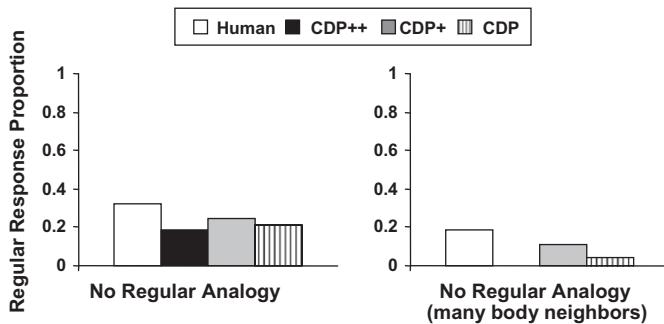
Andrews and Scarratt (1998) found that participants reading aloud nonwords with no regular analogy (i.e., nonwords that do not share orthographic bodies with any words that have a regular pronunciation) are far less likely to give regular responses than to nonwords that do share bodies with regular words. This no regular analogy effect was especially strong when the body of the nonwords was shared by many other words. As can be seen in Fig. D2, CDP++ (and indeed the whole CDP family) shows this effect (note that in the second experiment, CDP++ predicted no regular answers in the no regular analogy condition).

#### D.4. Position of irregularity

Rastle and Coltheart (1999) reported that the cost of irregularity was modulated by the position of the irregular correspondence in the word (but see Zorzi (2000) for an alternative account based on consistency). Words which had an irregular correspondence in an early position were slower to read aloud than words with an irregular correspondence in a late position. As can be seen from Fig. D3, this is also true of CDP++, with the first and second position irregular words showing a significant difference compared to their controls: first (10.9 cycles):  $t(38) = 3.29, SE = 3.30, p < .005$ ; second (6.1 cycles):  $t(74) = 3.27, SE = 1.85, p < .005$ , but not the third position irregular words (3.2 cycles,  $t(56) = 1.67, SE = 1.91, p = .10$ ). The quantitative performance of CDP++ was also good, explaining 29.39% of the variance. Due to potential confounds pointed out by Zorzi (2000) in the stimuli of Rastle and Coltheart set,

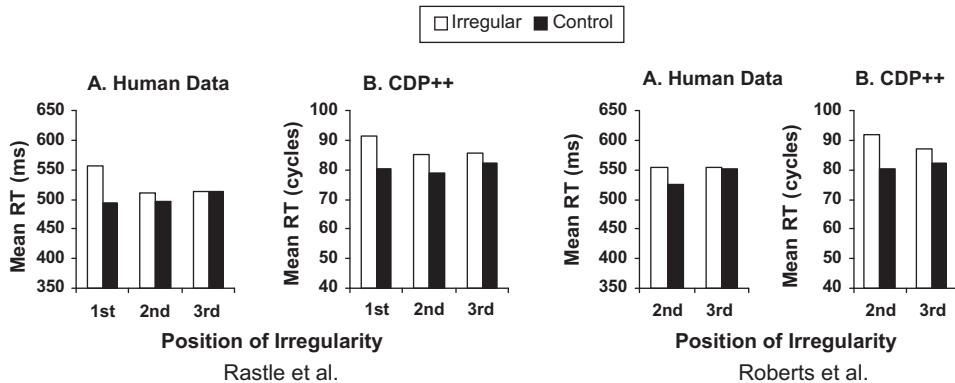
*Experiment 1**Experiment 2*

**Fig. D1.** Human data (milliseconds) and CDP++ simulations (cycles) of Jared's (2002) Experiment 1 and Experiment 2.



**Fig. D2.** Human data (response probabilities for regular pronunciations) and simulations of different models for the "no regular analogy nonwords" (Experiment 1) and the "no regular analogy with many body neighbors nonwords" (Experiment 2) of Andrews and Scarratt (1998).

Roberts, Rastle, Coltheart, and Besner (2003) ran a similar experiment to Rastle et al., examining second and third position regularity with supposedly better stimuli. They found that there was a much larger effect of second position irregularity than third position irregularity. CDP++ also predicted a similar pattern, displaying a significant position by regularity interaction,  $F(1, 100) = 4.05$ ,  $MSE = 246$ ,  $p < .05$ .



**Fig. D3.** Human data (milliseconds) and CDP++ simulations (cycles) of Rastle and Coltheart's (1998) and Roberts et al.'s (2003) irregularity by position interaction.

#### D.5. Body neighborhood

Apart from a significant length by lexicality interaction, which is also shown by CDP++,<sup>9</sup>  $F(3, 143) = 3.57$ ,  $MSE = 1044$ ,  $p < .05$ , Ziegler et al. reported that words and nonwords in a large body neighborhood are read aloud more quickly than those in a small body neighborhood. The 2.7 cycle difference shown by CDP++ between high and low body neighborhood items did not reach significance,  $F(1, 143) = 1.28$ ,  $MSE = 374$ ,  $p = .26$ . CDP++ also explained only 4.9% of the variance with the words and 5.2% with the nonwords, which is less than CDP+. However, on inspection of the data, there were a number of outliers. We therefore re-analyzed the data with a 2SD cutoff instead of a 3SD cutoff, which removed seven items. This increased the quantitative performance of the model (words: 10.5%; nonwords: 10.8%).

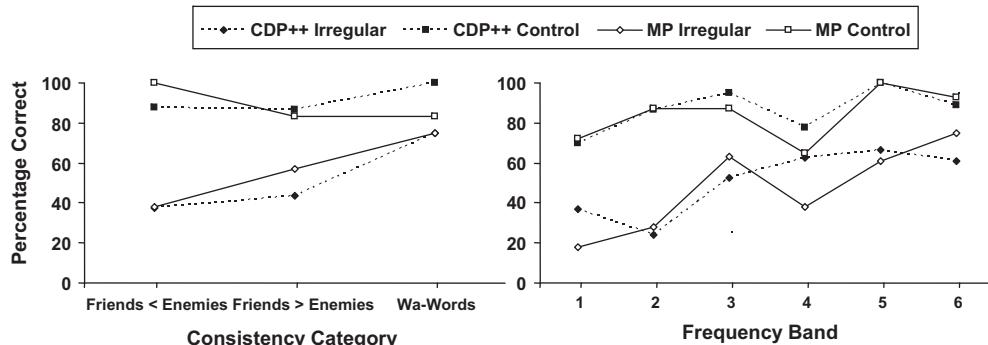
#### D.6. Pseudohomophone advantage

McCann and Besner (1987) found that people read aloud pseudohomophones faster than non-pseudohomophonic nonwords. CDP++ displayed the same pattern,  $t(128) = 2.52$ ,  $SE = 3.86$ ,  $p < .05$  (pseudohomophones: 111.7 cycles; nonwords: 121.5 cycles).

#### D.7. Surface dyslexia

MP is the most important acquired surface dyslexic studied using a single-case approach. Two important patterns she showed were a consistency effect, where the amount of errors she made on irregular words decreased as the consistency of the words increased (Patterson & Behrmann, 1997) and a frequency effect, where the amount of errors she made on irregular words decreased as the frequency of the words increased (Behrmann & Bub, 1992). We simulated MP in a similar way as in CDP+, where the inhibition parameters from the phonological lexicon to the phoneme output buffer was set to zero and the excitation parameter reduced (in this simulation, to 0.025). As with CDP+, we also increased the frequency scaling parameter (to 0.40), which makes low-frequency words comparatively more difficult to access in the lexicons than high-frequency words. Unlike CDP+, this simulation included both the monosyllabic and the disyllabic words that were used with MP. As can be seen in Fig. D4, the results showed that CDP++ produced a very similar pattern of results compared to MP with both data sets.

<sup>9</sup> Two nonwords that were actually words were run through the model as if they were nonwords.



**Fig. D4.** Performance of Surface Dyslexic MP and CDP++ on the stimuli of Patterson and Behrmann (1997; first graph) and Behrmann and Bub (1992; second graph). Frequency bands are those defined in Behrmann and Bub, with increasing numbers representing increasing frequency.

#### D.8. Phonological dyslexia

LB is a phonological dyslexic who displayed more errors on pseudohomophones that are orthographically similar to their base words (e.g., *meed/mead*) than orthographically dissimilar (e.g., *phocks/fox*). This was simulated with CDP+ by reducing the strength of all of the inhibitory connections in the model and reducing the strength of the sublexical activation. This same strategy was used to simulate LB with CDP++. The values of all of the inhibitory connections were reduced to  $\frac{1}{4}$  of their original strength and the sublexical network to phoneme output buffer activation parameter was reduced to .031. The results showed that, on the same stimuli set used in Coltheart et al. (2001), the percentage of words produced correctly by CDP++ was very similar to LB (LB/CDP++: Similar pseudohomophone, 85/72.5; Dissimilar pseudohomophone: 52/50; Nonword control group 1, 35/35; Nonword control group 2, 27/35).

#### D.9. Masked priming

Forster and Davis (1991) showed that if a masked prime word shared the same onset as a target word, then response times to the target were faster than if the prime was unrelated or if the prime shared the same rime as the target. We simulated this with CDP+ by simply running the prime word in the system for a small number of cycles before the target word.

The strategy used with CDP++ was slightly modified from CDP+. In particular, the parser was reset (i.e., the attentional window was moved back to the start of the string) only if the most active letter within the three letter window changed. In addition, after reset, it was assumed that the first letter in the parsing window was immediately available for use, as long as the letter was above the threshold that the parser uses to begin parsing.

With a 15 cycle prime duration, CDP++ produced RTs of 61.2 cycles for the onset primes, 63.2 cycles for the rime primes, and 63.6 cycles for the unrelated primes. Two *t*-tests showed that CDP++ read aloud targets preceded by onset primes faster than both rime ( $t(24) = 4.29, SE = .42, p < .001$ ) and unrelated ( $t(242) = 5.76, SE = .46, p < .001$ ) primes. Alternatively, target words preceded by rime and unrelated primes were not read aloud at significantly different speeds,  $t < 1$ .

#### Appendix E. Supplementary material

Supplementary data associated with this article can be found, in the on-line version, at doi:10.1016/j.cogpsych.2010.04.001.

## References

- Aiken, L. S., & West, S. G. (1991). *Multiple regression: Testing and interpreting interactions*. Newbury Park, London: Sage.
- Álvarez, C. J., Carreiras, M., & Taft, M. (2001). Syllables and morphemes: Contrasting frequency effects in Spanish. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28, 545–555.
- Andrews, S. (1989). Frequency and neighborhood effects on lexical access: Activation or search? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15, 802–814.
- Andrews, S., & Scaratt, D. R. (1998). Rule and analogy mechanisms in reading nonwords: Hough Dou Peaple Rede Gnew Wirds? *Journal of Experimental Psychology: Human Perception and Performance*, 53, 567–593.
- Ans, B., Carbonnel, S., & Valdois, S. (1998). A connectionist multiple-trace memory model for polysyllabic word reading. *Psychological Review*, 678–723.
- Arciuli, J., & Cupples, L. (2003). Effects of stress typicality during speeded grammatical classification. *Language and Speech*, 353–374.
- Arciuli, J., & Cupples, L. (2006). The processing of lexical stress in word recognition: Typicality effects and orthographic correlates. *The Quarterly Journal of Experimental Psychology*, 59, 920–948.
- Ashby, J., & Clifton, C. J. (2005). The prosodic property of lexical stress affects eye movements in silent reading. *Cognition*, 96, B89–B100.
- Baayen, R. H., Piepenbrock, R., van Rijn, H. (1993). The CELEX lexical database (CD-ROM): Linguistic Data Consortium, University of Pennsylvania.
- Balota, D., Cortese, M. J., Sergent-Marshall, D. S., Spieler, D. H., & Yap, M. J. (2004). Visual word recognition of single-syllable words. *Journal of Experimental Psychology: General*, 133, 283–316.
- 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.
- Balota, D. A., Yap, M. J., Cortese, M. J., Hutchison, K. A., Kessler, B., Loftis, B., et al (2007). The English lexicon project. *Behavior Research Methods*, 39, 445–459.
- Becker, S., Behrmann, M., Moscovitch, M., & Joordens, S. (1997). Long-term semantic priming: A computational account and empirical evidence. *Journal of Experimental Psychology: Learning Memory and Cognition*, 23, 1059–1082.
- Behrmann, M., & Bub, D. (1992). Surface dyslexia and dysgraphia: Dual routes, single lexicon. *Cognition Neuropsychology*, 9, 209–251.
- Besner, D., Twilley, R. S., McCann, R. S., & Seergobin, K. (1990). On the association between connectionism and data: Are a few words necessary? *Psychological Review*, 97, 432–446.
- Brown, P., Lupker, S. J., & Colombo, L. (1994). Interacting sources of information in word naming: A study of individual differences. *Journal of Experimental Psychology: Human Perception and Performance*, 20, 537–554.
- Bullinaria, J. A. (1997). Modelling reading, spelling and past tense learning with artificial neural networks. *Brain and Language*, 59, 236–266.
- Burzio, L. (1994). *Principles of English stress*. Cambridge: Cambridge University Press.
- Butler, B., & Hains, S. (1979). Individual differences in word recognition latency. *Memory and Cognition*, 7, 68–76.
- Chateau, D., & Jared, D. (2003). Spelling–sound consistency effects in disyllabic word naming. *Journal of Memory and Language*, 48, 255–280.
- Chen, H. C., & Vaid, J. (2007). Word frequency modulates the Basic Orthographic Syllabic Structure (BOSS) effect in English polysyllabic word recognition. *Language and Cognitive Processes*, 22, 58–82.
- Colombo, L. (1992). Lexical stress effect and its interaction with frequency in word pronunciation. *Journal of Experimental Psychology: Human Perception and Performance*, 18, 987–1003.
- 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.
- Content, A. (1991). The effect of spelling-to-sound regularity on naming in French. *Psychological Research*, 53, 3–12.
- Davis, C. J., & Bowers, J. S. (2006). Contrasting five different theories of letter position coding: Evidence from orthographic similarity effects. *Journal of Experimental Psychology: Human Perception and Performance*, 32, 535–557.
- Dupoux, E., Pallier, C., Sebastian-Galles, N., & Mehler, J. (1997). A distressing “deafness” in French? *Journal of Memory and Language*, 36, 406–421.
- Feldman-Stewart, D., & Mewhort, D. J. K. (1994). Learning in small connectionist networks does not generalize to large networks. *Psychological Research*, 56, 99–103.
- Ferrand, L. (2000). Reading aloud polysyllabic words and nonwords: The syllabic length effect reexamined. *Psychonomic Bulletin & Review*, 7, 142–148.
- Ferrand, L., & New, B. (2003). Syllabic length effects in visual word recognition and naming. *Acta Psychologica*, 113, 167–183.
- Flemming, E., & Johnson, S. (2007). Rosa's roses: Reduced vowels in American English. *Journal of the International Phonetic Association*, 37, 83–96.
- Forster, K. I., & Davis, C. (1991). The density constraint of form-priming in the naming task: Interference effects from a masked prime. *Journal of Memory and Language*, 30, 1–25.
- Fudge, E. (1984). *English word stress*. London: George Allen and Unwin.
- Garde, P. (1968). *L'Accent*. Paris: Presses Univ. France.
- Glushko, R. J. (1979). The organization and activation of orthographic knowledge in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance*, 5, 674–691.
- Gordon, M. K. (2006). *Syllable weight: Phonetics, phonology, typology*. New York: Routledge.
- Goswami, U. (2002). Phonology, reading development, and dyslexia: A cross-linguistic perspective. *Annals of Dyslexia*, 52, 139–163.
- Goswami, U., & Ziegler, J. C. (2006). A developmental perspective on the neural code for written words. *Trends in Cognitive Sciences*, 10, 142–143.

- Grainger, J., & Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review*, 103, 518–565.
- Gupta, P., & Touretzky, D. (1994). Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science*, 18, 1–50.
- Hall, T. A. (2002). Against extrasyllabic consonants in German and English. *Phonology*, 19, 33–75.
- Hall, T. A. (2006). English syllabification as the interaction of markedness constraints. *Studia Linguistica*, 60, 1–33.
- Halle, M., & Idsardi, W. J. (1997). r, Hypercorrection, and the elsewhere condition. In I. Roca (Ed.), *Derivations and constraints in phonology* (pp. 331–348). Oxford: Clarendon Press.
- Hammond, M. (1997). Vowel quantity and syllabification in English. *Language*, 73, 1–17.
- Hammond, M. (1999). *The phonology of English. A prosodic optimality-theoretic approach*. Oxford: Oxford University Press.
- Harm, M. W., & Seidenberg, M. S. (2004). Computing the meanings of words in reading: Cooperative division of labor between visual and phonological processes. *Psychological Review*, 111, 662–720.
- Harris, J. (1994). *English sound structure*. Oxford: Blackwell.
- Hayes, B. (1995). *Metrical stress theory: Principles and case studies*. University of Chicago Press.
- Heselwood, B. (2007). Schwa and the phonotactics of RP English. *Transactions of the Philological Society*, 105, 148–187.
- Heselwood, B. (2009). R vocalisation, linking R and intrusive R: Accounting for final schwa in RP English. *Transactions of the Philological Society*, 107, 66–97.
- Houghton, G., & Zorzi, M. (2003). Normal and impaired spelling in a connectionist dual-route architecture. *Cognitive Neuropsychology*, 20, 115–162.
- Hutzler, F., Ziegler, J. C., Perry, C., Wimmer, H., & Zorzi, M. (2004). Do current connectionist learning models account for reading development in different languages? *Cognition*, 91, 273–296.
- Jacobs, A. M., & Grainger, J. (1994). Models of visual word recognition: Sampling the state of the art. *Journal of Experimental Psychology: Human Perception and Performance*, 20, 1311–1334.
- Jared, D. (1997). Spelling–sound consistency affect the naming of high-frequency words. *Journal of Memory and Language*, 36, 505–529.
- Jared, D. (2002). Spelling–sound consistency and regularity effects in word naming. *Journal of Memory and Language*, 46, 723–750.
- Jared, D., & Seidenberg, M. S. (1990). Naming of multisyllabic words. *Journal of Experimental Psychology: Human Perception and Performance*, 16, 92–105.
- Jensen, J. T. (2000). Against ambisyllabicity. *Phonology*, 17, 187–235.
- Kawamoto, A. H., & Zembilidge, J. H. (1992). Pronunciation of homographs. *Journal of Memory and Language*, 31, 349–374.
- Kello, C. T. (2006). Considering the junction model of lexical processing. In S. Andrews (Ed.), *From inkmarks to ideas: Current issues in lexical processing*. New York: Taylor and Francis.
- Kelly, M. H. (2004). Word onset patterns and lexical stress in English. *Journal of Memory and Language*, 50, 231–244.
- Kelly, M. H., Morris, J., & Verrekia, L. (1998). Orthographic cues to lexical stress: Effects on naming and lexical decision. *Memory and Cognition*, 26, 822–832.
- Kessler, B., Treiman, R., & Mullenix, J. (2002). Phonetic biases in voice key response time measurements. *Journal of Memory and Language*, 47, 145–171.
- Ladefoged, P. (2001). *A course in phonetics* (4th ed.). Boston, MA: Heinle and Heinle.
- Lloyd, S., & Wenham, S. (2000). *The phonics handbook: In print letters (jolly phonics)*. Essex: Jolly Learning Ltd.
- Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28, 203–208.
- Lupker, S. J., Brown, P., & Colombo, L. (1997). Strategic control in a naming task: Changing routes or changing deadlines? *Journal of Experimental Psychology: Learning Memory and Cognition*, 23, 570–590.
- Lupker, S. J., Perea, M., & Davis, C. J. (2008). Transposed letter priming effects: Consonants, vowels and letter frequency. *Language and Cognitive Processes*, 23, 93–116.
- McCann, R., & Besner, D. (1987). Reading pseudohomophones: Implications for models of pronunciation assembly and the locus of word-frequency effects in naming. *Journal of Experimental Psychology: Human Perception and Performance*, 13, 14–24.
- McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: 1. An account of basic findings. *Psychological Review*, 88(5), 375–407.
- Miceli, G., & Caramazza, A. (1993). The assignment of word stress in oral reading: Evidence from a case of acquired dyslexia. *Cognitive Neuropsychology*, 10, 273–295.
- Monsell, S., Doyle, M. C., & Haggard, P. N. (1989). Effects of frequency on visual word recognition tasks: Where are they? *Journal of Experimental Psychology: General*, 118, 43–71.
- Mulatti, C., Reynolds, M. G., & Besner, D. (2006). Neighborhood effects in reading aloud: new findings and new challenges for computational models. *Journal of Experimental Psychology: Human Perception and Performance*, 32, 799–810.
- Paap, K. R., & Noel, R. W. (1991). Dual route models of print to sound: Still a good horse race. *Psychological Research*, 53, 13–24.
- Patterson, K., & Behrmann, M. (1997). Frequency and consistency effects in a pure surface dyslexic patient. *Journal of Experimental Psychology: Human Perception and Performance*, 23, 1217–1231.
- 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.
- Perry, C. (1999). Testing a computational account of category specific deficits. *Journal of Cognitive Neuroscience*, 11, 312–320.
- 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*. doi: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.
- Plaut, D. C. (1999). A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science*, 23, 543–568.
- 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.

- Rastle, K., & Coltheart, M. (1999). Serial and strategic effects in reading aloud. *Journal of Experimental Psychology: Human Perception and Performance*, 25, 482–503.
- Rastle, K., & Coltheart, M. (2000). Lexical and nonlexical print-to-sound translation of disyllabic words and nonwords. *Journal of Memory and Language*, 42, 342–364.
- Rastle, K., Havelka, J., Wydell, T. N., Coltheart, M., & Besner, D. (2009). Cross-script length effect: Further evidence challenging PDP models of reading aloud. *Journal of Experimental Psychology: Learning Memory and Cognition*, 35, 238–246.
- Rayner, K. (1998). Eye movements in reading and information processing: Twenty years of research. *Psychological Bulletin*, 124, 372–422.
- Rey, A., Courrieu, P., Schmidt-Weigand, F., & Jacobs, A. M. (2009). Item performance in visual word recognition. *Psychonomic Bulletin and Review*, 16, 600–608.
- Roach, P. (2000). *English phonetics and phonology* (3rd ed.). Cambridge: Cambridge University Press.
- Roberts, M. A., Rastle, K., Coltheart, M., & Besner, D. (2003). When parallel processing in visual word recognition is not enough: New evidence from naming. *Psychonomic Bulletin and Review*, 10, 405–414.
- Roelofs, A., & Meyer, A. S. (1998). Metrical structure in planning the production of spoken words. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24, 922–939.
- Seidenberg, M. S., & Waters, G. S. (1989). Word recognition and naming: A mega study [Abstract]. Paper presented at the Bulletin of the Psychonomic Society.
- Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, 96, 523–568.
- Seidenberg, M. S., Plaut, D. C., Petersen, A. S., McClelland, J., & McCrae, K. (1994). Nonword pronunciation and models of word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 20, 1177–1196.
- Selkirk, E. (1980). The role of prosodic categories in English word stress. *Linguistic Inquiry*, 11, 563–605.
- Ševa, N., Monaghan, P., & Arciuli, J. (2009). Stressing what is important: Orthographic cues and lexical stress assignment. *Journal of Neurolinguistics*, 22, 237–249.
- Share, D. L. (1995). Phonological recoding and self-teaching: Sine qua non of reading acquisition. *Cognition*, 55, 151–218.
- Sibley, D. E., Kello, C. T., & Seidenberg, M. S. (2010). A connectionist model of monosyllabic and bisyllabic naming. *European Journal of Cognitive Psychology*, 22(5). doi:10.1080/09541440903080583.
- Siegel, S., & Allan, L. G. (1996). The widespread influence of the Rescorla–Wagner model. *Psychonomic Bulletin and Review*, 3, 314–321.
- Spieler, D. H., & Balota, D. (1997). Bringing computational models of word naming down to the item level. *Psychological Science*, 8, 411–416.
- Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88, 135–170.
- Taft, M. (1979). Lexical access via an orthographic code: The Basic Orthographic Syllable Structure (BOSS). *Journal of Verbal Learning and Verbal Behavior*, 18, 21–39.
- Taft, M. (2001). Processing of orthographic structure by adults of different reading ability. *Language and Speech*, 44, 351–376.
- Taraban, R., & McClelland, J. L. (1987). Consistency effects in word recognition. *Journal of Memory and Language*, 26, 608–631.
- Treiman, R., Mullenix, J., Bijeljac-Babic, R., & Richmond-Welty, E. D. (1995). The special role of rimes in the description, use, and acquisition of English orthography. *Journal of Experimental Psychology: General*, 124, 107–136.
- Treiman, R., & Zukowski, A. (1996). Children's sensitivity to syllables, onsets, rimes, and phonemes. *Journal of Experimental Child Psychology*, 61, 193–215.
- van Oostendorp, M. (1998). Schwa in phonological theory. *GLOT*, 3, 5–9.
- Waese, M., & Jared, D. (2006). The role of intervocalic consonants in disyllabic word naming. Paper presented at the 47th Annual Meeting of the Psychonomic Society, Houston, Texas.
- Wagenmakers, E., Ratcliff, R., Gomez, P., & McKoon, G. (2008). A diffusion model account of criterion shifts in the lexical decision task. *Journal of Memory and Language*, 58, 140–159.
- Weekes, B. S. (1997). Differential effects of number of letters on words and nonword naming latency. *The Quarterly Journal of Experimental Psychology*, 50(A), 439–456.
- Widrow, G., & Hoff, M. E. (1960). Adaptive switching circuits. In *Institute of radio engineers, western electronic show and convention record, part 4* (pp. 96–104). New York: Institute of Radio Engineers.
- Yap, M. J., & Balota, D. A. (2009). Visual word recognition of multisyllabic words. *Journal of Memory and Language*, 60, 502–529.
- Yarkoni, T., Balota, D. A., & Yap, M. J. (2008). Moving beyond Coltheart's N: A New Measure of Orthographic similarity. *Psychonomic Bulletin & Review*, 15, 971–979.
- Yates, M. (2005). Phonological neighbors in visual word processing: Evidence from multiple tasks. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31, 1385–1397.
- Zevin, J., & Joanisse, M. (2000). Stress assignment in nonword reading. *Journal of Cognitive Neuroscience*, 41B, S5.
- Zevin, J. D., & Seidenberg, M. S. (2006). Simulating consistency effects and individual differences in nonword naming. *Journal of Memory and Language*, 54, 145–160.
- Ziegler, J. C., Besson, M., Jacobs, A. M., Nazir, T. A., & Carr, T. H. (1997). Word, pseudoword, and nonword processing: A multitask comparison using event-related brain potentials. *Journal of Cognitive Neuroscience*, 9, 758–775.
- Ziegler, J. C., Castel, C., Pech-Georgel, C., George, F., Alario, F.-X., & Perry, C. (2008). Developmental dyslexia and the dual route model of reading: Simulating individual differences and subtypes. *Cognition*, 107, 151–178.
- Ziegler, J., & Goswami, U. C. (2005). Reading acquisition, developmental dyslexia and skilled reading across languages: A psycholinguistic grain size theory. *Psychological Bulletin*, 131, 3–29.
- Ziegler, J. C., Perry, C., & Coltheart, M. (2003). Speed of lexical and nonlexical processing in French: The case of the regularity effect. *Psychonomic Bulletin and Review*, 10, 947–953.
- Ziegler, J. C., Perry, C., Jacobs, A. M., & Braun, M. (2001). Identical words are read differently in different languages. *Psychological Science*, 12, 379–384.
- Ziegler, J. C., Perry, C., & Zorzi, M. (2009). Additive and interactive effects of stimulus degradation: No challenge for CDP+. *Journal of Experimental Psychology: Learning Memory and Cognition*, 35, 306–311.

- Ziegler, J. C., Stone, G. O., & Jacobs, A. M. (1997). The feedback consistency effect in lexical decision and naming. *Behavior Research Methods, Instruments, & Computers*, 29, 600–618.
- Zorzi, M. (2000). Serial processing in reading aloud: No challenge for a parallel model. *Journal of Experimental Psychology: Human Perception and Performance*, 26, 847–856.
- Zorzi, M. (2010). The connectionist dual process (CDP) approach to modeling reading aloud. *The European Journal of Cognitive Psychology*, 22(5). doi:10.1080/09541440903435621.
- Zorzi, M., Houghton, G., & Butterworth, B. (1998a). The development of spelling–sound relationships in a model of phonological reading. *Language & Cognitive Processes*, 13, 337–371.
- Zorzi, M., Houghton, G., & Butterworth, B. (1998b). Two routes or one in reading aloud? A connectionist dual-process model. *Journal of Experimental Psychology: Human Perception and Performance*, 24, 1131–1161.