Behavior Research Methods, Instruments, & Computers
2004, 36 (3), 488-499

WordGen: A tool for word selection and nonword generation in Dutch, English, German, and French

WOUTER DUYCK, TIMOTHY DESMET, and LIEVEN P.C. VERBEKE
Ghent University, Ghent, Belgium

and

MARC BRYSBAERT
Royal Holloway, University of London, London, England

WordGen is an easy-to-use program that uses the CELEX and Lexique lexical databases for word selection and nonword generation in Dutch, English, German, and French. Items can be generated in these four languages, specifying any combination of seven linguistic constraints: number of letters, neighborhood size, frequency, summated position-nonspecific bigram frequency, minimum position-nonspecific bigram frequency, position-specific frequency of the initial and final bigram, and orthographic relatedness. The program also has a module to calculate the respective values of these variables for items that have already been constructed, either with the program or taken from earlier studies. Stimulus queries can be entered through WordGen's graphical user interface or by means of batch files. WordGen is especially useful for (1) Dutch and German item generation, because no such stimulus-selection tool exists for these languages, (2) the generation of nonwords for all four languages, because our program has some important advantages over previous nonword generation approaches, and (3) psycholinguistic experiments on bilingualism, because the possibility of using the same tool for different languages increases the cross-linguistic comparability of the generated item lists. WordGen is free and available at http://expsy.ugent.be/wordgen.htm.

One of the most important stages in psycholinguistic research on word processing is the construction of items. To be able to draw valid and general conclusions on the basis of an experiment's outcome, the selection of words has to be performed with the utmost care. Items have to be manipulated adequately on the experimental variables under scrutiny, and items in different conditions have to be matched appropriately on potentially confounding factors. Because this implies searching for unique and strict parameter combinations in huge databases, item construction is usually a time-consuming and laborious endeavor. Recently, there has been growing awareness that a high degree of care must be applied when constructing the nonword stimuli that are often used in psycholinguistic experiments (especially in the visual word recognition literature). Forster and Veres (1998), for example, showed that the masked orthographic priming effect (i.e., target words [e.g., *contrast*] are processed faster after tachistoscopic presentation of an orthographically related prime [e.g., *contract*]), interacted with the word-likeness of the nonword stimuli, which merely served as distractors in the experiment.

This article presents WordGen, an easy-to-use tool that can substantially simplify and speed up the laborious job of item construction and checking, which has mostly been done manually up to the present day. WordGen uses the CELEX[1] database (Baayen, Piepenbrock, & van Rijn, 1993, 1995) and the Lexique[2] database (New, Pallier, Brysbaert, & Ferrand, 2004) to generate word and nonword items in Dutch, English, German, and French. The program is free and available at http://expsy.ugent.be/wordgen.htm. In order to install the program, the 1993 or 1995 CD-ROM version of the CELEX lexical database is needed. Upon installation, the CELEX lemma frequency databases of Dutch, English, and German are read from this CD-ROM and parsed for future use with WordGen. Because the Lexique database is freely available (http://www.lexique.org) and distributed under a GNU license, the data needed for French word and nonword generation are included in the program's download, so it is not necessary to download the Lexique database separately.

Before going into the details of the program and the underlying algorithms, we will briefly discuss the linguistic variables that can be controlled for by the program and their importance in the psycholinguistic literature. These variables include word frequency, neighborhood size, bigram frequency, orthographic relatedness, and word length.

This research was made possible by the Research Foundation–Flanders (F.W.O.–Vlaanderen, Belgium), of which W.D. and T.D. are postdoctoral fellows. We thank Susan Dunlap, Koen Mertens, Lael Schooler, Natasha Tokowicz, Jonathan Vaughan, and an anonymous reviewer for excellent comments on an earlier draft of this article. Correspondence should be addressed to W. Duyck, Department of Experimental Psychology, Ghent University, Henri Dunantlaan 2, B-9000 Ghent, Belgium (e-mail: wouter.duyck@ugent.be).

One of the most important linguistic variables in word recognition is word frequency: Words that occur more frequently are processed faster and more accurately than words that occur less frequently. This effect was first demonstrated in tachistoscopic recognition (Howes & Solomon, 1951) and was later generalized to a wide range of tasks, including lexical decision (e.g., Whaley, 1978) and word naming (e.g., Forster & Chambers, 1973). It is important to control for word frequency in psycholinguistic experiments because this variable has subtle effects, emerging not only between highly frequent and highly infrequent words, but even between frequent and slightly less frequent words. In the mid-1990s, the suggestion was made that all frequency effects in the literature were actually confounded age-of-acquisition effects (Morrison & Ellis, 1995). The age of acquisition of a word is the age at which it is first learned (Carroll & White, 1973; Gilhooly, 1984). The hypothesis was that more frequent words are processed faster not because they are more frequent, but because they are generally acquired earlier. However, at present it seems that both frequency and age of acquisition have independent effects in word processing (e.g., Bonin, Chalard, Meot, & Fayol, 2001; Brysbaert, Lange, Van Wijnendaele, 2000; Gerhand & Barry, 1998; Izura & Ellis, 2004; Morrison & Ellis, 1995). Word frequency is still controlled or manipulated in virtually all word processing studies.

Another variable that affects word processing is orthographic neighborhood size. The neighborhood size of an item is the number of existing words that can be obtained by changing one letter of that item (Coltheart, Davelaar, Jonasson, & Besner, 1977). For instance, the English word *song* has six orthographic neighbors: *long*, *sung*, *pong*, *gong*, *sing*, and *tong*. A large neighborhood size enhances the performance on naming and lexical decision, especially for low-frequency words (Andrews, 1989; Grainger, 1990; McCann & Besner, 1987). In nonword items, neighborhood size could be an indicator of how wordlike a nonword is. For instance, an unpronounceable nonword such as *hzva* has no orthographic neighbors in English, whereas a more pronounceable nonword such as *dith* has 3 neighbors, and a pseudoword such as *pank* has 15 neighbors. As will become clear later, WordGen applies this observation in order to create pronounceable and very wordlike nonwords.

Another lexical variable that our program allows to constrain is type bigram frequency. Bigrams are the adjacent letter pairs of an item. For instance, the word *code* has three bigrams: *co*, *od*, and *de*. The effect of bigram frequency on word processing is a bit controversial. For instance, early effects of bigram frequency on word recognition (e.g., Rice & Robinson, 1975; Rumelhart & Siple, 1974) were later argued to be confounded effects of subjective familiarity (e.g., Gernsbacher, 1984). Also, some recent studies fail to find an effect of bigram frequency (e.g., Andrews, 1992), whereas others do find an effect (e.g., Westbury & Buchanan, 2002). Nevertheless, bigram frequency is still controlled for in numerous recent psycholinguistic studies (e.g., Bertram & Hyönä, 2003; Locker, Simpson, & Yates, 2003; Martensen, Maris, & Dijkstra, 2003; Yates, Locker, & Simpson, 2003). Moreover, from the perspective of this program, it is also an interesting variable to consider when making nonword items because on average the higher the summated bigram frequency of a nonword, the more wordlike it is.

By allowing to indicate which letters should and should not be part of the generated items, WordGen also allows for the manipulation of the orthographic overlap between items. Numerous studies have found that orthographically related items can prime each other (e.g., Brysbaert, 2001; Grainger & Ferrand, 1996; van Heuven, Dijkstra, Grainger, & Schriefers, 2001). For example, recognition of the target word *contrast* is faster when it is preceded by a tachistoscopically presented prime such as *contract*, than by a control prime that has no letters in common with the target word. Of course, with WordGen the orthographic overlap cannot only be manipulated, but it can also be controlled for, which is of crucial importance in experiments that are exploring the independent effects of phonological or semantic priming. Suppose, for example, that a control prime is needed for the semantically related prime–target pair *mouse—cheese*. In this case, it is possible to probe WordGen for a control prime having *se* as the last two letters, so that any semantic priming effect cannot be attributed to the fact that these two letters overlap between the experimental prime and target. Interestingly, orthography not only influences visual word recognition processes, but has also been shown to play an important role in speech production (e.g., Damian & Bowers, 2003) and speech perception (e.g., Miller & Swick, 2003; Slowiaczek, Soltano, Wieting, & Bishop, 2003). Hence, this WordGen feature may also be useful for such studies.

Finally, our program also allows for the constraining of the length of a word or nonword by indicating the number of letters. It has been shown that longer (non)words have longer lexical decision and naming times (e.g., Chumbley & Balota, 1984; Forster & Chambers, 1973; Weekes, 1997; Whaley, 1978). Virtually all word processing experiments control for length.

In the following sections, we will discuss how these variables have been implemented in WordGen. We will subsequently deal with the four different panes of the program: (1) options, (2) generation, (3) checking, and (4) batch mode. For detailed technical information about features of these four panes that are not discussed below, we refer readers to the WordGen manual, which is available on the WordGen Web site and in the program itself.

Options

Before looking up word or nonword information in the "checking" pane or creating items in the "generation" pane, some options can be set. First, one of four languages needs to be selected: Dutch, English, German, or French. Next, WordGen allows for the output to be saved to a data file. If this option is not chosen, the output only appears in the window on the right side of the program and is lost when the program is shut down. The user can also choose for the program to provide detailed output.

When this option is selected, the user gets a list of all of an item's neighbors and the frequency of all of its separate bigrams. Next, the nonword searching time can be limited to any number of seconds (and is set to 30 sec as a default). This option is provided because asking the program for a nonword with a constraint combination that is too narrowly defined or even impossible can lead to an infinite (or very long-lasting) search. For instance, asking the program for a Dutch 10-letter nonword with 14 neighbors and a very low summated bigram frequency is unlikely to be successfully completed within a reasonable time (if it is possible to find such a nonword at all). Thus, the program will continue searching if the search time is not set to a specific limit. Users who are looking for nonwords that have to meet certain strict—but not impossible—constraints should set the time limit very high or should deactivate it entirely. In practice, the to-be-generated nonwords will be matched to existing words, mostly to reasonable combinations of constraints. Note that this time limit does not apply to word generation because it does not take much time for the program to perform an exhaustive check of every word in the databases against the constraints that were set.

Generation

To generate a word, the program registers the values of the constraints that were set by the user. The program randomly selects an entry in the CELEX or Lexique database and starts a serial search through the database looking for the first word that satisfies the combination of constraints provided by the user. If the end of the database is reached, WordGen restarts from the beginning of the database and runs until the point of entry. Each time the user asks to generate a new word (even using the same parameter settings during the same session) a different random entry in the database is selected. If "linear search mode" is selected in the "options" pane, WordGen will always return the word that occurs first in the alphabetically sorted database.

To generate a nonword, the program assembles a string of randomly selected letters and verifies whether the letter string is an existing word in the lexical database for that particular language. Next, every constraint is checked, and as soon as one of them is violated the random letter string is rejected and the process starts all over again until a letter string is assembled that conforms to all constraints or until the time limit that was set in the "options" pane is reached. The latter case might be an indication that the parameters were set too narrowly and that the constraints should be broadened.

In practice, a psycholinguist in the process of constructing a nonword often bases his or her nonword on an existing word and changes one letter to turn it into a nonword. This heuristic ensures that nonwords are mostly reasonably wordlike. The program can be set to use this approach, but we included the other (random letter generation) strategy as well because we believe it is desirable to allow for as much variation as possible in the type of nonwords. For instance, we did not want to exclude possible nonwords that had no neighbors but are still pronounceable wordlike letter strings (e.g., "syspor"), a possibility that is excluded when basing nonwords on existing words. Of course, we provided some other search options to ensure the wordlikeness of the nonwords generated by the program.

When generating a word or nonword, seven constraints can be set. The first and most straightforward constraint is the number of letters the generated item should have.

The second constraint is neighborhood size, or the number of orthographic neighbors an item can have. If this option is set, the program checks which words in the respective CELEX/Lexique database have all letters but one in common with the candidate word/nonword. In this way, a highly accurate count of the neighborhood size for a (non)word in a given language is obtained. This is especially useful for Dutch and German, for which no neighborhood size norms are available at present. Hence, the program allows avoidance of more elaborate and less accurate assessment strategies of neighborhood size, which are often used in studies in these languages, such as asking a number of independent participants to name as many neighbors as possible of the items that will be used in the experiment (e.g., van Hell & Dijkstra, 2002). When setting the neighborhood size constraint during a word/nonword search, it is important to know that neighborhood size is related to word length. For instance, whereas almost all 3-letter words have at least a couple of neighbors, longer words mostly have zero, one, or two neighbors. Figure 1 shows the distribution of neighborhood size as a function of language and word length.

This information might be useful when setting the neighborhood constraint. For example, looking at Figure 1, it clearly would not make much sense to ask the program for a Dutch 8-letter word with 12 neighbors. It should be noted that in the figures we left out the number of words with zero or one neighbors. Including this information would have distorted the scale of the *y*-axis too much because a huge amount of words have fewer than two neighbors. Note that this information would not be very useful anyway, given the aim of these histograms, because searching for an item with fewer than two neighbors is never an unreasonable constraint.

The third constraint that can be set is the word frequency of an item. Obviously, this constraint can only be set in word generation. In our program, the frequency of words is based on the lemma frequencies provided in the CELEX database for Dutch, English, and German and the lemma frequencies provided in the Lexique database for French. This implies that the written word frequency of the word *book*, for example, includes the frequency of occurrence in the corpus of the wordforms related to the noun (e.g., *book*, *books*) and the wordforms related to the verb (e.g., [to] *book*, *booked*, . . .). We decided to use lemma—and not, for example, wordform—frequencies for a number of reasons. First and most importantly, the former is by far most often used in psycholinguistic re-

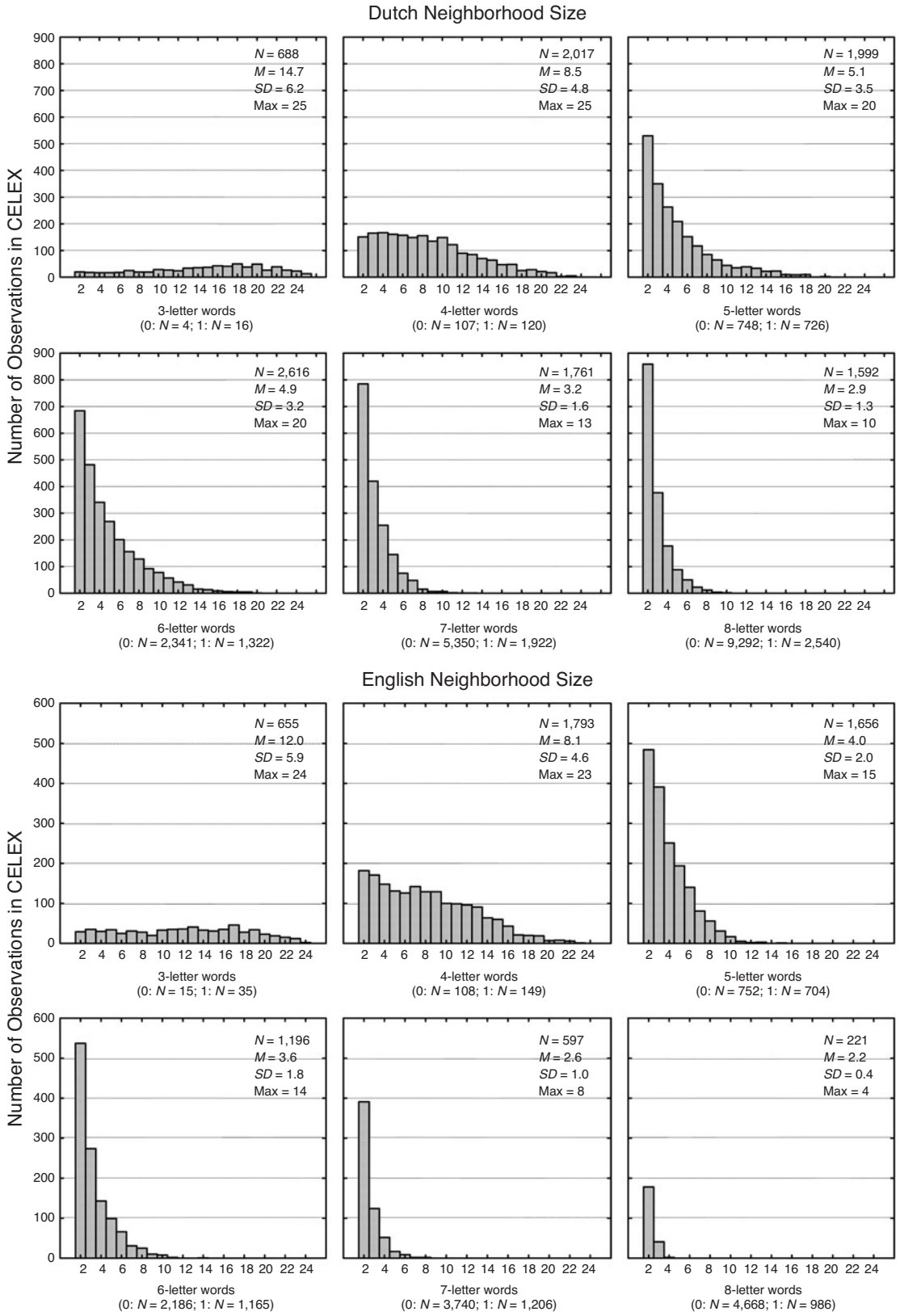

Figure 1. Neighborhood size: Number of occurrences in the CELEX/Lexique databases as a function of language and word length. Words having 0 or 1 neighbor are omitted to prevent *Y* scale distortion. Respective *N*s for these words are indicated between brackets below the graphs.

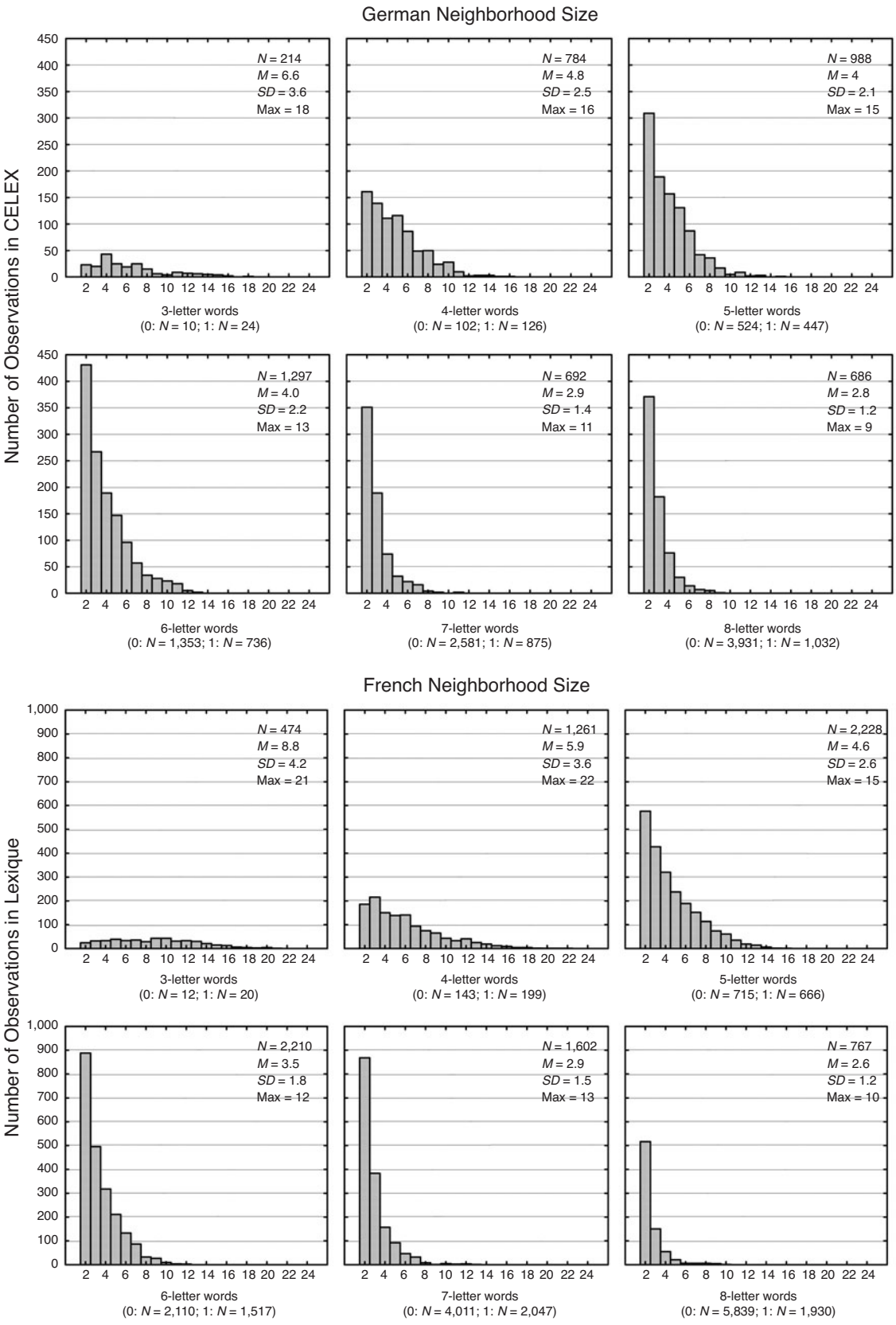

Figure 1 (Continued).

search. Second, the lemma database is smaller, which substantially speeds up the search process in the database (especially important for nonword generation). Third, due to extensive manual coding and disambiguation, the lemma database is more transparent with respect to its records than the wordform database. For example, the wordform database contains a lot of compound entries consisting of several words (e.g., *go back on*). Any program resolving these issues (as WordGen is only able to process words, not word groups) is basically repeating part of the lemma coding. Fourth, due to the lemma database's considerable size, it is likely that the variables of interest to WordGen, calculated on the basis of the lemma database, would correlate substantially with those based on the wordform database. Finally, several studies (e.g., Baayen, Dijkstra, & Schreuder, 1997; New, Brysbaert, Segui, Ferrand, & Rastle, 2004) suggest that the processing of words is partly driven by the frequency of morphologically related wordforms (e.g., plurals), which favors the lemma frequency approach (because these wordforms are grouped in the same lemma entry).

In order to ensure high comparability between different languages and studies, WordGen uses a relative measure of lemma frequency, that is, frequency per million words in the corpus. This recognizes that the databases of each language contain a different amount of words. Also, WordGen primarily operates on the logarithm of the lemma frequency per million words. This corrects for the fact that the difference between a frequency of 3 and 5 is more important than the difference between a frequency of 103 and 105. Because logarithmic values are sometimes hard to interpret, WordGen is also able to report plain frequencies per million. However, because these are calculated by inverting the original logarithmic values and are therefore approximate measures, we advise use of the latter.

The fourth constraint is summated type bigram frequency. Our program summates the position-nonspecific frequency of each bigram of a (non)word, based on how many times a bigram occurs in the CELEX or Lexique database independent of its position in the word. For example, the Dutch word *boek* has a bigram frequency of 19,898, which is the sum of the number of occurrences of each of its bigrams: *bo* (4,123), *oe* (9,120), and *ek* (6,655) in the CELEX. Because the four languages have a different number of words in the database, there is a big difference between the bigram frequencies for these languages. For instance, the Dutch and English databases in the CELEX contain 124,136 and 52,447 entries, respectively. This means that, on average, Dutch summated bigram frequencies will be more than twice as high as English summated bigram frequencies.[3] Also, because the program works with summated bigram frequencies, on average the bigram frequency for short words will be lower than the bigram frequency for long words. To help the user set the summated bigram frequency constraint, we included a figure with the distribution information of bigram frequencies as a function of language and word length.

Again, these figures should make clear that it does not make much sense, for instance, to ask for a Dutch four-letter word with a summated bigram frequency of at least 80,000. As an aid to the user, the program adapts the default constraints for summated bigram frequency as a function of the language and the number of letters that was chosen. However, depending on the needs of the user it is advisable to look at the histograms in Figure 2 to narrow the range of this constraint. As for nonword generation, using higher levels of summated bigram frequency will generally result in more wordlike nonwords.

In addition to the summated bigram frequency constraints, the minimum "legal" bigram frequency and the minimum position-specific onset/suffix bigram frequency can be set. These two constraints were added to increase the efficiency of constructing plausible nonwords. If only the summated bigram frequency were constrained, it is possible that the program would generate a nonword only one of whose bigrams would be highly frequent in a given language (leading to a high summated bigram frequency) while the other bigrams were highly infrequent, leading to an unpronounceable nonword. The minimum legal bigram frequency allows indication of what the minimum bigram frequency should be for any of the bigrams of an item, so the user can make sure that the nonword does not contain any infrequent bigrams that do not appear in any word in the respective lexical database. In practice, the default values set in the program have proven to be adequate, and the onset and suffix position-specific bigram frequency can be constrained. This is because bigrams that are very frequent in some places in a word can still be very infrequent as the first or last two letters of the word, so that many randomly generated nonwords can appear to be unusable. For instance, the bigram *rt* is quite frequent in English (it occurs 1,266 times in the lemma corpus), but it is never the first bigram of a word. The position-specific onset/suffix bigram frequency constraint makes sure that both the onset bigram and the suffix bigram occur a certain number of times as the onset or the suffix of a word. Hence, while the program includes the possibility of generating nonpronounceable nonwords, we strongly advise searching for parameter settings of these constraints which are adequate to the stimuli at hand, in order to obtain pronounceable nonwords.

In addition to these bigram frequency constraints, we included the possibility of using a widely adopted heuristic to enhance nonword generation even further (especially for nonwords longer than 7 letters). When using the heuristic, the program randomly selects an existing word and then exchanges one random letter for another one (irrespective of whether it is a vowel or a consonant) to turn it into a nonword that conforms to the other constraints that were set. This leads to very wordlike nonwords. Together with the bigram frequency constraints, this heuristic ensures the generation of nonwords that vary widely

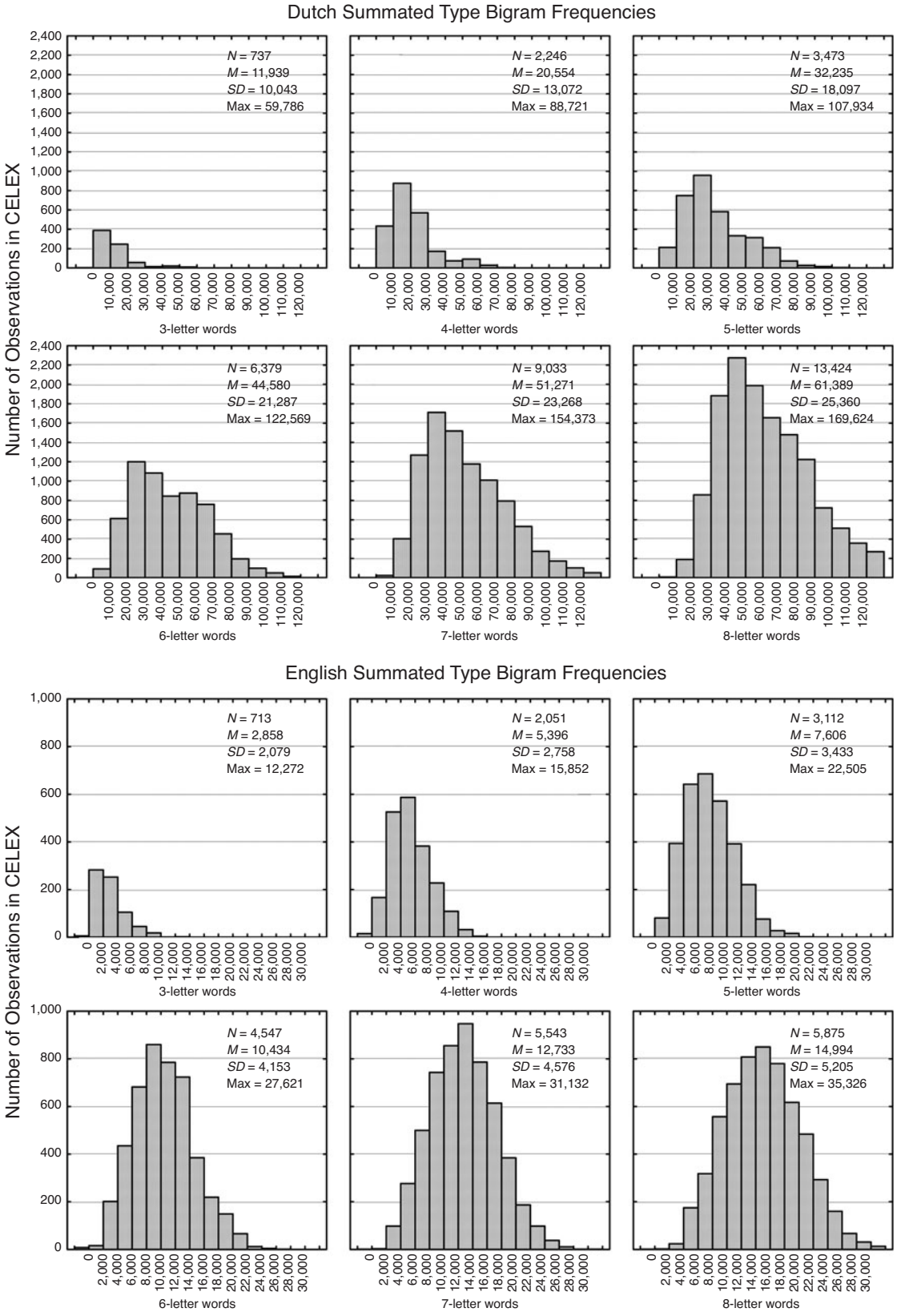

Figure 2. Summated type bigram frequency: Number of occurrences in the CELEX/Lexique databases as a function of language and word length. Notice that the graphs have different bigram frequency scales for different languages, due to the fact that WordGen uses four corpora that consist of a strongly differing number of words.

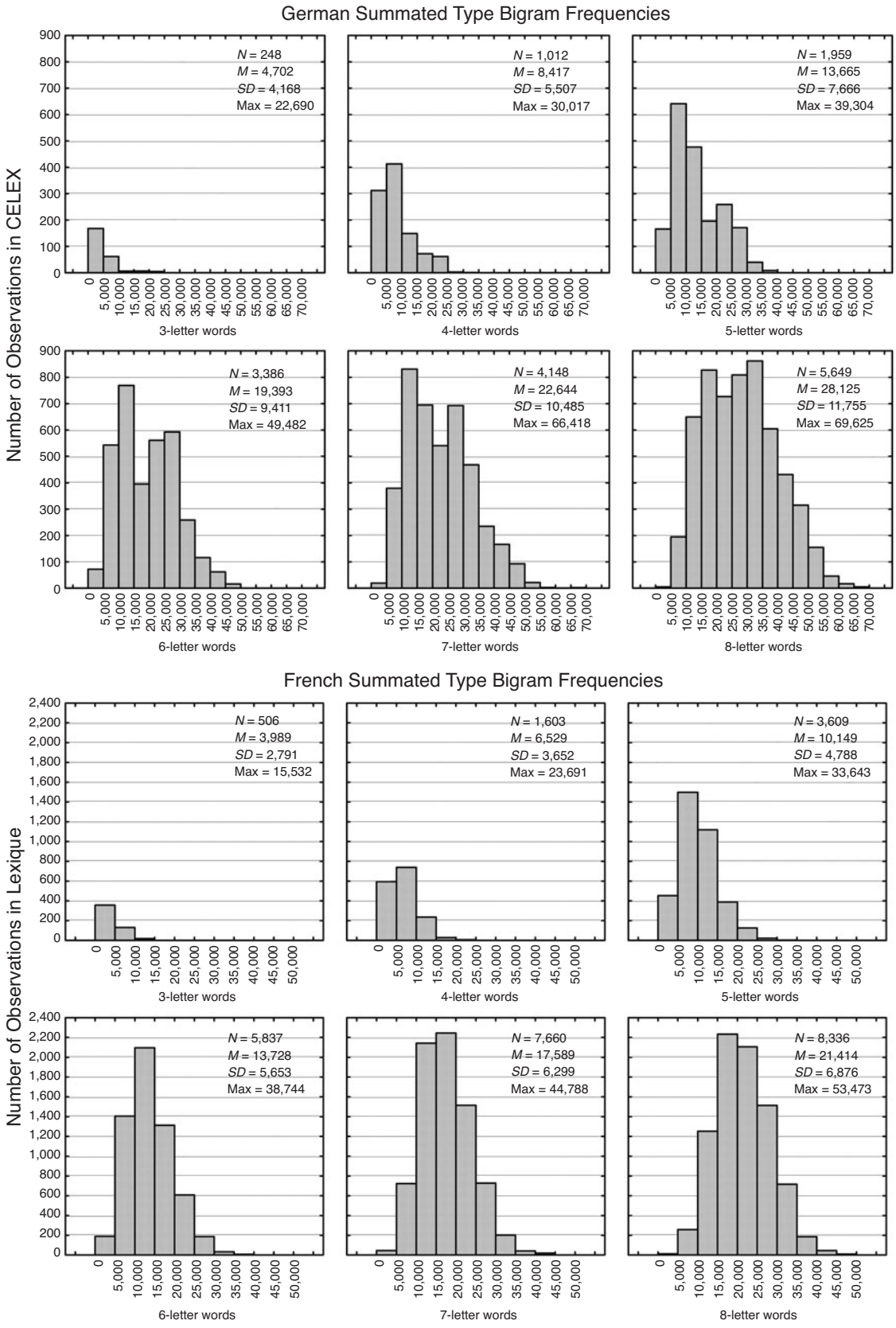

Figure 2 (Continued).

between very wordlike nonwords to completely unpronounceable nonwords.

As an illustration of how the different constraint settings influence the nature of the generated nonwords, we ran a series of tests with different parameter settings. We generated 4-, 5-, 6-, 7-, and 8-letter nonwords either (1) with no constraints set at all, (2) with the minimum legal bigram frequency set at 30 and the minimum position-specific bigram frequency set at 15, (3) with the latter two constraints and the number of neighbors set to 1, or (4) using the heuristic without any other constraints set. For each of these conditions, we let the program generate 100 Dutch nonwords (for each of the different numbers of letters) and counted how many were pronounceable.

The results of this test are presented in Table 1. It is obvious from this table that using the minimum legal bigram frequency and the position-specific bigram frequency greatly improves the quality of the nonwords compared to when no constraints are set (McNemar $\chi^2 = 206.005$, $p < .001$). This is especially true when it is also requested that the nonwords should have one neighbor (McNemar $\chi^2 = 274.004$, $p < .001$). With these constraints, the ratio of pronounceable nonwords is about 80%, which is quite high given the fact that the underlying algorithm only uses orthographic information and does not have an extensive set of complicated grapheme–phoneme conversion rules. Hence, probing the program two times for a nonword will almost always result in a pronounceable nonword satisfying a combination of several lexical constraints. We believe that this is a considerable improvement over classical nonword generation, which is often done manually (and therefore much slower) or by pseudo-automation, without a clearly defined set of lexical criteria used to generate these items. This underspecification of nonword characteristics often makes it very difficult to compare nonword items across studies. This is especially troubling, given the fact that changing the nature of filler nonwords can influence the processing of the word stimuli, which is the actual object of interest (e.g., see Forster & Veres, 1998, in which it was shown that the wordlikeness of used nonword targets interacts with the orthographic masked priming effect).

Table 1
Number of Pronounceable Dutch Nonwords (out of 100) as a Function of Number of Letters and Constraint Settings

	Constraints			
Number of Letters	No Constraints[a]	Bigram[b]	Bigram + Neighbor[c]	Heuristic[d]
4	19	73	80	68
5	8	55	84	62
6	6	49	77	60
7	12	43	80	66
8	7	40	X[e]	74

[a]None of the constraints were set to a specific value.
[b]The minimum legal bigram frequency was set to 30; the minimum legal position-specific bigram frequency was set to 15.
[c]The minimum legal bigram frequency was set to 30, the minimum legal position-specific bigram frequency was set to 15, and the number of neighbors was set to 1.
[d]Only the heuristic was used; no other constraints were set.
[e]Because random nonword generation takes very long in this condition (there are about 2×10^{11} possible random 8-letter string combinations), we advise considering the heuristic approach for nonwords longer than 7 letters.

The next constraint is the possibility of using a wildcard. This option allows the user to indicate whether the item should contain a specific letter in a specific position. For instance, a search for a 5-letter word with a "p" in the second letter position can be indicated by typing *p*** next to the "use wildcard" option; a search for a 7-letter word with an "a" in the third position and an "s" in the fifth position can be asked for by **a*s**.

The final option is the forbidden letter list, which offers the possibility to indicate which letters should not be part of the generated item. If multiple letters need to be excluded, they should be typed next to each other without blank spaces or commas. For instance, when a word is needed that should not contain the letters *m* and *r*, the user should type *mr* next to the "forbidden letter list" option.

When generating nonwords or selecting word stimuli, it is often the case that researchers need several words/nonwords satisfying the same constraints. Also, somebody may wish to see a list of several nonwords, all satisfying specified constraints, before manually selecting one from that list. In those cases, we advise the use of "generate list" feature in the bottom frame of the "generation" pane. With this option, WordGen generates a list of words/nonwords satisfying the same set of parameters and prints it to a file. That way, for example, it is possible to ask WordGen to generate a list of 100 English nonwords using a single click, instead of stimulus by stimulus.

Checking

In addition to the generation of words and nonwords, the program also allows calculation of the respective values of the variables mentioned above for already constructed lists of words or nonwords, either created with WordGen itself, or as a control of the stimuli of earlier studies. When checking an item, the program verifies whether it is a word or a nonword by seeing whether it can be found in the CELEX or Lexique database. When the checked item is a word, the (log) frequency per million, the number of neighbors, and the summated type bigram frequency are provided. The same is true when the checked item is a nonword apart from the fact that the log frequency is not provided.

WordGen also allows cross-language checking of (non) words. With this feature, (non)words are simultaneously parsed through the different lexical databases associated with the selected languages. This feature enables one, for example, to easily retrieve the language-specific frequency of cross-lingual homographs (i.e., words that are orthographically identical but have a different meaning in the other language, e.g., *room*, meaning *cream* in Dutch). It is also possible to quickly determine the Dutch neighborhood size of English nonwords, which may be useful for studies focusing on language-independent activation of lexical knowledge. Van Heuven, Dijkstra, and Grainger (1998), for example, showed that the recognition of En-

glish target words in Dutch–English bilinguals is influenced by the Dutch neighborhood size of those words. RTs were longer for English words having many of Dutch neighbors.

Batch Mode

Although WordGen is designed to provide an easy-to-use "click-and-retrieve" graphical user interface for word selection and nonword generation, repetitive queries can be highly automated with the batch mode feature. This allows the experienced user to specify the different parameter settings of a large stimulus set before WordGen is probed for results. As a result, WordGen can be programmed to search independently and uninterrupted for a large stimulus set (even of a whole experiment), without human intervention. Commands may be entered in the command line box, or through separate batch files, which can be created with a simple text editor. The syntax for this batch mode is described in the WordGen manual.

Contributions to the Field

In the psycholinguistic literature, a number of tools and databases are available for stimulus generation. This is especially true for English and French, but less so for Dutch and German. In this section, we will give a concise overview of the most frequently used tools that are available for each of the four languages, and we will outline the extra contribution of WordGen for each of these languages.

In English, there is the MRC psycholinguistic database (Coltheart, 1981), which contains a large number of lexical properties of words, such as number of syllables, word frequency, imageability, age of acquisition, part of speech, stress pattern, and so forth (see http://www.psy.uwa.edu.au/mrcdatabase/uwa_mrc.htm). For the construction of nonword items, there is the ARC nonword database, which contains monosyllabic nonword items that conform to English phonological rules (Rastle, Harrington, & Coltheart, 2002; see also http://www.maccs.mq.edu.au/~nwdb/). We believe that our tool is complementary to both the MRC and ARC. For instance, the MRC Psycholinguistic Database does not contain summated bigram frequencies, nor does it provide neighborhood size ratings. The ARC, on the other hand, does not contain multisyllabic nonwords, and it only contains pseudohomophones or very wordlike nonwords, thus not allowing for as much variety in nonwords as WordGen does. Hence, we believe that WordGen may be useful (1) to calculate neighborhood size and bigram frequency measures of English words and nonwords and (2) for the construction of English multisyllabic or low wordlike nonwords.

In French, there is the freely available Lexique database (New et al., in press), which contains a huge amount of French lexical information, and Lexop, a computerized lexical database which provides measures of the relationship between phonology and orthography for French monosyllabic words (Peereman & Content, 1999). Again, we think that WordGen is an interesting extension to the French situation. First, no other tool is available for nonword generation in French. Second, the availability of a number of different types of bigram frequency makes our tool very helpful for French stimulus generation.

Finally, WordGen is especially suited to be used for Dutch and German psycholinguistic experiments because these languages lack publicly available databases, similar to those for English and French mentioned above, which contain frequently used lexical measures such as neighborhood size, bigram frequencies, and functions like nonword generation. For instance, there are no available norms of neighborhood size for Dutch, forcing researchers to resort to inaccurate methods of controlling for neighborhood size, such as asking participants to name as many neighbors of the items that will be used in the experiment. Now, WordGen provides more accurate norms, which can also be searched for by multiple entry points. It is not only possible to check how many neighbors a (non)word has but also to ask the program for a (non)word that has a specific number of neighbors. This advantage also holds for English and French, for which norms exist for words, but where it is harder to find words that have a prespecified number of neighbors (especially in combination with other lexical constraints).

Besides increasing the possibility to generate words and nonwords in Dutch, English, German, and French, WordGen has some other advantages. First, this program is ideally suited for stimulus generation in the fast-growing research domain of bilingualism. The same program and norms can be used to construct items in different languages, enhancing the comparability of the item lists over languages. This is especially true given that the combination of different lexical variables can be constrained at the same time. Until now, item construction for studies on bilingualism usually relied on databases and norms that differ between languages and studies, which made it difficult to directly compare the stimuli of studies yielding conflicting results.

A final advantage of this program is that it allows for a great variation in nonword items, ranging from highly recognizable nonwords to pseudowords. The way nonwords are created traditionally—by taking a word and changing one letter—does not easily allow for the manipulation of wordlikeness (although this heuristic is also available in WordGen). This variation in wordlikeness is possible in WordGen by the specific way in which the nonwords are constructed (creating random letter strings), which does not artificially exclude nonwords that have no neighbors and are very nonwordlike. Moreover, the possibility of specifying bigram frequency and number of neighbors is a big advantage for researchers interested in the influence of nonword characteristics on performance in word recognition tasks (e.g., Forster & Veres, 1998).

Future Extensions of WordGen

Several extensions of the program may be useful features for the future. Most important, the program can eas-

ily be updated to include new languages. Because WordGen only needs orthographic (and frequency) information for nonword generation (and word selection), new languages can—and will—easily be added to the program. The only thing that is needed to include a new language is a reliable list of lemmata and their frequency. For example, Spanish may easily be incorporated with the LexEsp corpus. Second, it may be worthwhile to add other variables to the program, such as word class, imageability, familiarity, or age of acquisition. This, however, calls for the collection of large sets of norms for different languages. Third, the quality of the nonwords may be further improved by the inclusion of trigrams, or even *n*-grams. Also, more measures of position-specific bigram frequency may be included. Finally, whereas this is clearly beyond the scope of the current WordGen program, a similar program including not only orthographic, but also phonological and morphological information (e.g., to determine word body neighbors) could certainly be very useful.

REFERENCES

Andrews, S. (1989). Frequency and neighborhood effects on lexical access: Activation or search? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **15**, 802-814.

Andrews, S. (1992). Frequency and neighborhood effects on lexical access: Lexical similarity or orthographic redundancy? *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **18**, 234-254.

Baayen, R. H., Dijkstra, T. & Schreuder, R. (1997). Singulars and plurals in Dutch: Evidence for a parallel dual-route model. *Journal of Memory & Language*, **37**, 94-117.

Baayen, R. H., Piepenbrock, R., & van Rijn, H. (1993). *The CELEX lexical data base* [CD-ROM]*:* Philadelphia: Linguistic Data Consortium, University of Pennsylvania.

Baayen, R. H., Piepenbrock, R., & van Rijn, H. (1995). *The CELEX lexical data base* [CD-ROM 2nd Release]*:* Philadelphia: Linguistic Data Consortium, University of Pennsylvania.

Bertram, R., & Hyönä, J. (2003). The length of a complex word modifies the role of morphological structure: Evidence from eye movements when reading short and long Finnish compounds. *Journal of Memory & Language*, **48**, 615-634.

Bonin, P., Chalard, M., Meot, A., & Fayol, M. (2001). Age-of-acquisition and word frequency in the lexical decision task: Further evidence from the French language. *Cahiers de Psychologie Cognitive/Current Psychology of Cognition*, **20**, 401-443.

Brysbaert, M. (2001). Prelexical phonological coding of visual words in Dutch: Automatic after all. *Memory & Cognition*, **29**, 765-773.

Brysbaert, M., Lange, M., & Van Wijnendaele, I. (2000). The effects of age-of-acquisition and frequency-of-occurrence in visual word recognition: Further evidence from the Dutch language. *European Journal of Cognitive Psychology*, **12**, 65-85.

Carroll, J. B., & White, M. N. (1973). Age of acquisition norms for 220 picturable nouns. *Journal of Verbal Learning & Verbal Behavior*, **12**, 563-576.

Chumbley, J. I., & Balota, D. A. (1984). A word's meaning affects the decision in lexical decision. *Memory & Cognition*, **12**, 590-606.

Coltheart, M. (1981). The MRC psycholinguistic database. *Quarterly Journal of Experimental Psychology*, **33(A)**, 497-505.

Coltheart, M., Davelaar, E., Jonasson, J. T., & Besner, D. (1977). Access to the internal lexicon. In S. Dornic (Ed.), *Attention and performance VI* (pp. 535-555). London: Academic Press.

Damian, M. F., & Bowers, J. S. (2003). Effects of orthography on speech production in a form-preparation paradigm. *Journal of Memory & Language*, **49**, 119-132.

Forster, K. I., & Chambers, S. M. (1973). Lexical access and naming time. *Journal of Verbal Learning & Verbal Behavior*, **12**, 627-635.

Forster, K. I., & Veres, C. (1998). The prime lexicality effect: Form-priming as a function of prime-awareness, lexical status, and discrimination difficulty. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **24**, 498-514.

Gerhand, S., & Barry, C. (1998). Word frequency effects in oral reading are not merely age-of-acquisition effects in disguise. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **24**, 267-283.

Gernsbacher, M. A. (1984). Resolving 20 years of inconsistent interactions between lexical familiarity and orthography, concreteness, and polysemy. *Journal of Experimental Psychology: General*, **113**, 256-281.

Gilhooly, K. J. (1984). Word age-of-acquisition and residence time in lexical memory as factors in word naming. *Current Psychological Research*, **3**, 24-31.

Grainger, J. (1990). Word-frequency and neighborhood frequency-effects in lexical decision and naming. *Journal of Memory & Language*, **29**, 228-244.

Grainger, J., & Ferrand, L. (1996). Masked orthographic and phonological priming in visual word recognition and naming: Cross-task comparisons. *Journal of Memory & Language*, **35**, 623-647.

Howes, D. H., & Solomon, R. L. (1951). Visual duration threshold as a function of word probability. *Journal of Experimental Psychology*, **41**, 401-410.

Izura, C., & Ellis, A. W. (2004). Age of acquisition effects in translation judgement tasks. *Journal of Memory & Language*, **50**, 165-181.

Locker, L., Jr., Simpson, G. B., & Yates, M. (2003). Semantic neighborhood effects on the recognition of ambiguous words. *Memory & Cognition*, **31**, 505-515.

Martensen, H., Maris, E., & Dijkstra, T. (2003). Phonological ambiguity and context sensitivity: On sublexical clustering in visual word recognition. *Journal of Memory & Language*, **49**, 375-395.

McCann, R. S., & 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 & Performance*, **13**, 14-24.

Miller, K., & Swick, M. D. (2003). Orthography influences the perception of speech in alexic patients. *Journal of Cognitive Neuroscience*, **15**, 981-990.

Morrison, C. M., & Ellis, A. W. (1995). Roles of word-frequency and age of acquisition in word naming and lexical decision. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **21**, 116-133.

New, B., Brysbaert, M., Segui, J., Ferrand, L., & Rastle, K. (2004). The processing of singular and plural nouns in French and English. *Journal of Memory & Language*, **51**, 568-585.

New, B., Pallier, C., Brysbaert, M., & Ferrand, L. (2004). *Lexique 2*: A new French lexical database. *Behavior Research Methods, Instruments, & Computers*, **36**, 516-524.

Peereman, R., & Content, A. (1999). LEXOP: A lexical database providing orthography–phonology statistics for French monosyllabic words. *Behavior Research Methods, Instruments, & Computers*, **31**, 376-379.

Rastle, K., Harrington, J., & Coltheart, M. (2002). 358,534 nonwords: The ARC nonword database. *Quarterly Journal of Experimental Psychology*, **55A**, 1339-1362.

Rice, G. A., & Robinson, D. O. (1975). The role of bigram frequency in the perception of words and nonwords. *Memory & Cognition*, **3**, 513-518.

Rumelhart, D. E., & Siple, P. (1974). Process of recognizing tachistoscopically presented words. *Psychological Review*, **81**, 99-118.

Slowiaczek, L. M., Soltano, E. G., Wieting, S. J., & Bishop, K. L. (2003). An investigation of phonology and orthography in spoken-word recognition. *Quarterly Journal of Experimental Psychology*, **56A**, 233-262.

van Hell, J. G., & Dijkstra, T. (2002). Foreign language knowledge can influence native language performance in exclusively native contexts. *Psychonomic Bulletin & Review*, **9**, 780-789.

van Heuven, W. J. B., Dijkstra, T., & Grainger, J. (1998). Orthographic neighborhood effects in bilingual word recognition. *Journal of Memory & Language*, **39**, 458-483.

van Heuven, W. J. B., Dijkstra, T., Grainger, J., & Schriefers, H. (2001). Shared neighborhood effects in masked orthographic priming. *Psychonomic Bulletin & Review*, **8**, 96-101.

Weekes, B. S. (1997). Differential effects of number of letters on word and nonword naming latency. *Quarterly Journal of Experimental Psychology*, **50A**, 439-456.

Westbury, C., & Buchanan, L. (2002). The probability of the least likely non–length-controlled bigram affects lexical decision reaction times. *Brain & Language*, **81**, 66-78.

Whaley, C. P. (1978). Word–nonword classification time. *Journal of Verbal Learning & Verbal Behavior*, **17**, 143-154.

Yates, M., Locker, L. Jr., & Simpson, G. B. (2003). Semantic and phonological influences on the processing of words and pseudohomophones. *Memory & Cognition*, **31**, 856-866.

NOTES

1. The CELEX lexical database, among other things, contains frequency information about the lemmata (including different word classes) of three different languages: Dutch, English, and German. The Dutch corpus consists of 124,136 lemmata compiled from written sources of every kind (containing 42 million words). The English corpus consists of approximately 46,000 lemmata that were extracted from the COBUILD corpus, which contains almost 18 million words, mostly from written sources of many kinds. WordGen only operates on those written frequency measures. The German corpus consists of approximately 51,000 lemmata, mostly originating from various written texts (containing 5.5 million words). For more detailed information about CELEX, we refer to Baayen et al. (1993, 1995).

2. The Lexique database, among other things, contains frequency information for about 55,000 French lemmata, compiled from the Frantext database, which consists of approximately 31 million words from various written sources. For more detailed information about Lexique, we refer to New et al. (in press).

3. In order to make cross-language comparisons of summated bigram frequency, one could consider transforming the obtained measures in accordance with the number of lemmata in the respective databases.

(Manuscript received December 22, 2003;
revision accepted for publication July 19, 2004.)