Jemma König, Andreea S. Calude, and Averil Coxhead

Using character-grams to automatically generate pseudowords and how to evaluate them

Abstract: This paper provides a practical solution to the problem of generating (good) pseudowords, which are commonly used in vocabulary testing and experimental research in applied linguistics, and introduces an empirically-founded solution to evaluating the suitability of pseudowords for different tasks. In the first part of the paper we propose a novel way of generating pseudowords – a character-gram chaining algorithm. A major advantage of the algorithm is that it does not require any knowledge of the language, thereby facilitating the generation of pseudowords in any language. Secondly, there is currently a lack of formal criteria for evaluating pseudowords, both in terms of (i) their orthographic fit in the target language they are intended for, and (ii) their suitability for use in various lexical processing and language teaching tasks. In the second part of the paper, we argue for the need to evaluate pseudowords, propose a set of linguistic criteria for evaluating the generated pseudowords, and provide a comparison with other current pseudoword lists using this criteria.

Keywords: pseudowords, automatic generation, evaluation, corpus linguistics

Introduction

The concept of creating and using pseudowords is not new. Applied linguists, linguists, and indirectly, language teachers, have been using pseudowords in lexical decision tasks and vocabulary tests for a very long time. As such, there are several existing techniques for generating pseudowords, and several databases and software applications that utilise these techniques. However, there are also, of course, some limitations to the existing techniques and the software that use them. Primarily, their reliance on existing knowledge and understanding of the language, such as knowledge of syllabification, sub-syllabification, or simply knowing which letters can be inserted or deleted from a word while still resulting in a phonetically or orthographically legal form.

Furthermore, there does not appear to be any empirical evidence confirming the quality of the pseudowords being created, to the best of our knowledge. Generating *good* pseudowords is an act of fine-balance, because on the one hand, the generated forms must, of course, not be existing words, while at the same time, they need to be sufficiently close in resembling possible words already in existence (yet not so close that they could be mistaken for real words in a testing situation, we return to this point later). Quality of pseudowords, in this context, could refer to both (i) the *legal* structure of pseudowords – conforming to language internal rules, and to (ii) their *suitably* for use in various lexical tasks – such as whether they are more suited to morphology experiments, vocabulary testing, incidental vocabulary learning experiments, and so on.

A pseudoword (nonword, imaginary word, or disguised form) is a unit of text or speech that has no real meaning in a language but has linguistically appropriate orthographic and phonological structure (Nordquist 2018). It has the form of a word and is spelled in a predictable way, but does not exist in the lexicon (Groff 2003). Pseudowords play an essential role in lexical processing research and language teaching. Pseudowords are used by linguistics researchers to test language production processes. In morphology experiments, for example, the most famous pseudoword of all, *wug,* is used to test plural rule productivity (Berko 1958). Berko demonstrated the implicit knowledge of linguistic morphology in children. Her research used pseudowords to evaluate children's knowledge of morphological rules, proving that children were capable of forming suitable endings, producing plurals, possessives, and past tense. Pseudowords are also utilised in phonetic decoding (Cardenas 2009), measuring pronunciation latency in learners (Schwartz 2013) and visual word recognition through lexical decision tasks and naming tasks (Balota et al. 2007). Applied linguists and language testers have used pseudowords to assess the credibility of learner's responses in non-native vocabulary tests. Meara's (2010) English as a Foreign Language (EFL) vocabulary test incorporates pseudowords into the assessment of learner's vocabulary size, using them to judge a learners' self-evaluation of vocabulary knowledge. Another large scale example comes from Keuleers, Stevens, Mandera and Brysbaert (2015), who used Wuggy (Keuleers and Brysbaert, 2010) to create 20,653 nonwords (along with 52,849 words) to include in a large-scale online vocabulary size study in Dutch. Keuleers et al. (2015) shuffled the final list of 73,500 nonwords and words and broke them into sublists of 100 items which were then used to create a yes/no test. Psycholinguists and Applied Linguists (e.g. (Arndt and Woore 2018; Elgort

and Warren 2014; Saragi et al. 1978; Webb 2005, 2007) have used pseudowords to assess factors in incidental lexical learning, such as the extent to which vocabulary learning can happen informally, including what and how much is retained (phonological information, grammatical function and semantic content) and how much exposure is required for such learning to take place.

In this paper, we are not wishing to enter into a debate on the merits or otherwise of pseudowords in research. Instead, we first consider existing pseudoword creation techniques and the systems that use these techniques, as well as the limitations of both. Then we introduce a character-gram chaining algorithm – our newly proposed method, a set of criteria for evaluating the legality of pseudowords, and suitability considerations which researchers may want to use when choosing them. Finally, we consider the usefulness of this new approach for use with languages other than English and in different domains.

Existing techniques and systems for generating pseudowords

Three main techniques are currently used to generate pseudowords: (i) manipulating a stimulus, (ii) using high frequency bi-grams, and (iii) combining sub-syllabic elements. The first technique, manipulating a stimulus, involves a stimulus word (a word that exists in the lexicon) and manipulates it in some way to create a pseudoword (see Table 1 for the stimulus *pilot*). The stimulus can be altered by changing one or two characters, either by insertion, deletion, transposition, or replacement. Alternatively, a composite pseudoword can be created by adding a prefix or suffix to the stimulus, as long as it does not form a word that exists in the lexicon (R Harald Baayen and Schreuder 2011). The English Lexicon Project (ELP) (Balota et al. 2007) is a behavioural database of 40,481 stimulus words, contrived from the Kučera and Francis norms (1967) and the CELEX database (R H Baayen et al. 1993), and 40,481pseudowords that were created by changing 1 or 2 characters in each of the stimulus words, alternating the location of the manipulated characters between words.

[TABLE 1 NEAR HERE]

The second technique, using high frequency bi-grams, involves combining two letter sequences (bi-grams) that appear together frequently to form pseudowords. Programs such as WordGen (Duyck et al. 2004), that use this technique, tend to also consider neighbourhood size and orthographic relatedness. Table 2 shows an example for the pseudoword *reroin*. WordGen uses bigram frequency, in-part, for pseudowords generation in Dutch, English, German, and French. It generates a random sequence of letters which is validated as a pseudoword dependent on a set of seven constraints: number of letters, neighbourhood size, frequency, summated bigram frequency, minimum bigram frequency, initial bigram frequency, final bigram frequency, and orthographic relatedness (Duyck et al. 2004). Depending on the language, WordGen uses either the CELEX or Lexique (New et al. 2004) as its lexicon. A random selection of letters are considered to be a pseudoword only if it is not an existing word in the lexicon, and it meets all seven criteria.

[TABLE 2 NEAR HERE]

The third technique for creating pseudowords involves combining sub-syllabic elements within a language by breaking existing syllables down into their sub-syllabic elements and then joining them back together. A syllable is a unit of sound, typically made up of a nucleus (usually a vowel) and an optional onset (initial sound) and coda (final sound). This approach takes legal sub-syllabic elements (onset, nucleus, and coda) from existing words and combines them to form a pseudoword (Keuleers and Brysbaert 2010). Table 3 shows an example for the pseudoword *shib.* The ARC Nonword Database is an example of a system which creates pseudowords by joining sub-syllabic elements. The database holds a collection of over 350,000 monosyllabic non-words which combine onsets with rhymes (nucleus and coda) from sound relationships derived from the CELEX database. Another system that combines sub-syllabic elements is Wuggy (Keuleers and Brysbaert 2010). Given a list of syllabified words, it segments each word into sub-syllabic elements and builds a tree of all possible legal sub-syllabic combinations. The tree is then traversed to retrieve all possible pseudowords. Wuggy uses five lexicons (CELEX, Lexique, E-HITZ, B-PAL, and the Frequency Dictionary of Contemporary Serbian Language) to support pseudoword generation in seven languages: Dutch, English, German, French, Spanish, Serbian, and Basque.

[TABLE 3 NEAR HERE]

Problems with existing methods for creating pseudowords

Each of the existing techniques and systems for creating pseudowords comes with its own set of limitations, as will be discussed in this section in turn. Broadly speaking, these limitations have to do with one or several of the following

problems: (1) in order to create a list of pseudowords, the user needs ample knowledge of the language for which pseudowords are being created, for instance, in-depth knowledge about syllable structure rules of that language, (2) only a handful of the major languages are represented, (3) there is no way of evaluating whether the pseudowords created could actually pass as legal words of that language or whether they might be suitable for a given task.

Manipulating a stimulus requires knowledge of which characters can be inserted, deleted, or transposed while still resulting in a legal pseudoword. Using high frequency bi-grams requires knowledge of which combinations are legal within the language. Combining sub-syllabic elements requires knowledge of syllabification, and if phonetic syllables are used (as in the ARC Nonword Database) then understanding conversion from phonotactic to orthographic forms is also required.

Having a high number of possible pseudoword combinations is a risk when combining letter sequences or sub-syllabic elements. In the case of combining sub-syllabic elements, the number of combinations increases exponentially; monosyllabic words have hundreds of thousands of possible combinations, while polysyllabic words have billions (Keuleers and Brysbaert 2010). As a solution to high combination possibilities, software like WordGen and Wuggy provide building constraints or search criteria which restrict the pseudowords that are returned, for example, number of neighbours, word frequency and summated bigram frequency. This makes searching for pseudowords much more achievable, but can result in more complex software applications that can be confusing for researchers to interact with.

Two lexicons tend to be most widely used for generating pseudowords: CELEX and Lexique. CELEX (R H Baayen et al. 1993) is a lexical database that contains information on orthography, phonology, morphology, syntax, and word frequency for words in English, German, and Dutch. Lexique (New et al. 2004) is a lexical database for French language which contains information on gender, number, grammatical category, and word frequency. It is used by both the applications that support French: WordGen and Wuggy. Both CELEX and Lexique are general-purpose lexicons, meaning that they generate general-purpose pseudowords. Support for domain-specific pseudowords appears to be lacking, which could be problematic in cases where second language learners are being tested on their knowledge of academic or discipline-specific vocabulary (e.g. from Coxhead's Academic Word List (2000), or from disciplines such as biological sciences or engineering). Pseudowords, in these instances, would need to resemble Graeco-Latin words and be possibly longer than general purpose pseudowords, to make the test more realistic for learners.

The lexicons that are used by pseudoword generating software also limit the language support that is provided. ARC and ELP support English alone, WordGen supports four languages including English, and Wuggy supports seven. Wuggy also has the capacity to be extended to support any alphabet-based language. However, extending Wuggy to support other languages requires a list of syllabified words in the desired language, and information about how the syllables are segmented. There is, therefore, a need to develop a system that can be applied to a wider range of languages without requiring the in-depth knowledge of each (such as syllabification and sub-syllabification).

Each of the four existing pseudoword generating applications have some form of criteria that determines how pseudowords are created. However, to our knowledge, none has any formal criteria to evaluate the legal structure or suitability of pseudowords post-production. Each application has a different approach to generating pseudowords, and different criteria (grammars, or principles) for restricting the forms of the generated pseudowords. Perhaps, more importantly, they have differing views, even if only slightly, on what constitutes a *legal* pseudoword. The ARC database focuses on phonological principles (allowing illegal bigrams and orthographic onsets and bodies), while Wuggy focuses on orthographic forms (Keuleers and Brysbaert 2010: 629).

Furthermore, although each approach mentions the importance of suitable pseudowords, in terms of their use in lexical processing, incidental language learning, and lexical decision tasks, to our knowledge there is no evidence of their *suitability* having been tested or evaluated. A lack of suitability could have implications for how useful the pseudowords are and for the generalisation of the results found in such studies. The main problem is that different tasks have different suitability requirements. For instance, in incidental lexical learning experiments, it is desirable that pseudowords closely resemble existing simplex words, but in language testing situations, it may be undesirable to have only simplex words (depending on the type of vocabulary being tested, simplex words may stand out from the rest of the words used), and also, forms which resemble existing words too closely could be mistaken as real words under time pressure, which may lead to an incorrect penalty for the L2 learner being tested.

So while *legality* concerns a type of criteria that is overall largely desirable in all applications of pseudowords, *suitability* considerations relate to requirements which will by their very nature vary across tasks. Therefore, we purport that a set of pseudoword evaluation considerations are beneficial to the field, with regard to both *legality* and *suitability*.

Towards some solutions

The present study proposes a new approach to creating pseudowords that is not susceptible to any of the limitations of existing approaches, and explores ways of conducting post-creation evaluations on pseudowords. This study has two contributions:

1. An automated pseudoword generation technique that can be extended cross-linguistically.
2. An introduction to novel formal pseudoword evaluation techniques, both in terms of their *legal* form and *suitability* for various lexical tasks.

We start by introducing our character-gram-chaining algorithm, which is a computationally simple approach to generating pseudowords. Next we describe a set of evaluation criteria that we designed to evaluate the legal form of pseudowords, and some possible considerations for assessing their suitability in various lexical tasks. Finally, we demonstrate how the character-gram chaining algorithm has been designed to work with any alphabet-based language, without requiring any knowledge of that language, and how it can be used to create domain or language-specific pseudowords, based solely on an input wordlist.[i]

CGCA: character-gram chaining algorithm

In 1990, Bell, Cleary, & Witten discussed using statistical analysis of n-gram frequencies to model natural language. An early example of how this might be done can be found in Miller and Selfridge (1950). Bell et al. (1990: 80-81) suggested that "frequencies of n-grams can be used to construct order *n*-1 models". They descibe the model as an order *n*-1 model, where the first *n*-1 characters of an n-gram are used to predict the *n*th character. The examples in Table 4 demonstrate the first 100 characters that Bell, Cleary, & Witten (1990) generated using natural language modelling with different sized n-gram models. As shown in the table, although the 12-gram model is not perfect, the resemblence to natural language improves noticeably each time the n-gram size increases. Bell, Cleary, & Witten (1990) used n-gram models to reconstruct sections of text within a language, but their work has lead to our questioning whether a similar technique can be used to construct individual pseudowords.

[TABLE 4 NEAR HERE]

Both the n-gram model by Bell, Cleary, & Witten, and the sub-syllabic pseudoword technique used by Keuleers & Brysbaert (2010) and Rastle, Harrington, & Coltheart (2002) have motivated us to create a Character-Gram-Chaining Algorithm (CGCA) to generate pseudowords. Figure 1 gives an overview of the algorithm in four modularised steps while Appendix A gives the full details.

[FIGURE 1 NEAR HERE]

The CGCA algorithm can accept either a wordlist or corpus as input. First, it extracts all unique tokens from the input, creating the *origin wordlist*[ii] that is used to generate pseudowords. Next, it extracts all possible character-grams from the origin wordlist, and finally, it iteratively generates and validates each chain of character-grams, resulting in a list of pseudowords specific to the origin that was used to generate them. This allows researchers to generate either general-purpose or domain-specific pseudowords, based on the input that they pass through to our algorithm.

The CGCA can generate pseudowords from an input that has as little as 100 unique words, to large general-purpose corpora or wordlists such as the Range programme lists (Nation et al. 2002) or Nation's (2012) British National Corpus and Corpus of Contemporary American English (BNC/COCA) lists. However, the number of pseudowords that can be generated is dependent on the character-gram size used, for example, a list of 100 unique words can be used to generate 100 pseudowords when 2-grams or 3-grams are used, but only half as many using 4-grams.

We used CGCA and the BNC/COCA lists (Nation, 2012) to create pseudowords of various n-gram size. Table 5 gives the first 10 such pseudowords for 2-grams, 3-grams, 5-grams, 8-grams. We also generated pseudowords using a combination of n-gram lengths (r-grams). The reason that we can have a maximum size of 8-grams is that for larger numbers than 8, there would not be enough character-grams extracted to chain together to generate pseudowords.

[TABLE 5 NEAR HERE]

Designing post-production evaluation criteria and requirements

We propose that once generated, pseudowords should undergo evaluation. We argue that the evaluation process needs to involve two separate types of criteria, namely legality and suitability. Determining whether a pseudoword conforms

to the rules of a language, and determining whether that pseudoword is suitable for a particular type of lexical task, require two different types of evaluation criteria.

Legal elements are those that we can prove to be legal, in terms of the character-combinations that exist in the language (in a wordlist, lexicon or corpus), rather than all legal elements within the language. Elements that we measure as not legal are not necessarily illegal in the language. They simply cannot be proven to be legal in a sub-collection of the language, in an origin wordlist for example. Our approach to pseudoword generation makes use of the premise that chaining legal character-grams together will sometimes result in a legal pseudoword. Of course, there are other restrictions on the process of building a well-formed word, such as which sequences are being chained and their position in the word. These restrictions will thus lead to some ill-formed generated pseudowords. In particular, using bi-grams (character-grams of size 2) results in legal character sequences of size 2, but when chained together, these bi-grams can result in larger unseen or potentially illegal character combinations, which George Bernard Shaw's well-known example GHOTI illustrates.[iii] We have designed a set of three criteria that should be used to evaluate the legal form of pseudowords for English, outlined in Table 6.

[TABLE 6 NEAR HERE]

If a sequence of characters appears in a pseudoword but does not appear in a real word (from the origin wordlist), then the pseudoword cannot be *proven* legal (it might still conform to the word formation rules of the target language, but given that no words in the original wordlist contain the sequence, we cannot be sure whether it does or not). Conversely, if all the character sequences that appear in a pseudoword also appear in real words in the origin wordlist, then the pseudoword can be said to conform to the rules of the target language. This is what we hope for when assessing the legality of a given pseudoword.

Our version of legality is not too strict in that it is flexible enough to allow words which may break phonotactic constraints of a given language, for example, *un* + *table* would be classified as legal using the above criteria, even though *un* would never attach to a noun in English. However, as one anonymous reviewer points out, the legality criteria could be tightened much further to adhere to phonotactic constraints, but this comes with disadvantages in that there will be fewer pseudowords accepted and more language-specific knowledge required to evaluate legality of this type.

Measuring the *suitability* of a given pseudoword is not to do with whether or not a form could be a legitimate word in a given language, but rather, it has to do with its suitability for use. In lexical decision tasks, for example, the more dissimilar a pseudoword is to a word, the faster the reaction time (Keuleers and Brysbaert 2010). Incidental lexical learning experiments require shorter words with at most one productive affix, and which appear to be similar to existing forms (Webb 2007). Vocabulary tests (Meara 2010) and identification-as-retrieval tasks (Rueckl and Olds 1993) require pseudowords that do not have their own existing inferred meaning. Decoding-tasks use pseudowords of varying difficulty (Proença et al. 2017). The primary importance for suitability appears to be whether the pseudowords that are being generated are too similar to existing words, not similar enough, or within the range of what is required for lexical tasks. We propose four considerations for evaluating the suitability of pseudowords (English-specific), as given in Table 7.

[TABLE 7 NEAR HERE]

Using these considerations, each pseudoword could be given a binary score of either 0 or 1 for each of the above criteria. These scores could then be used to determine whether a particular pseudoword is (or is not) suited towards a particular lexical task. For example: for lexical decision tasks, a pseudoword that is only one character away from a real word, such as "dauntings" (a rushed or slightly absent-minded participant may not even hear or see the plural "s" and assume the form is a real word, "daunting"), may produce different reaction times than one that is not; pseudowords that score 1 in the polymorphic category (such as "unbehave") may be more suited to incidental lexical learning experiments; and pseudowords that score 0 in the polymorphic category (such as "hydraft") may be more suited to vocabulary tests and identification-as-retrieval tasks as they do not include a real root and affix that meaning can be inferred from.

While the CGCA method is cross-linguistically applicable, evaluating the generated pseudowords needs to be done according to language-specific means. That is, applying the rules of each target language separately and constructing suitability criteria separately for each language, as we have done above for English.

We applied each of the criteria outlined above to the first 100 pseudowords generated using each n-gram size (from 2-8). Table 8 shows the results of the *individual legal evaluation*, which was done automatically using a Python script. When the CGCA pseudowords were evaluated against their origin wordlist (Nation's BNC/COCA lists), 12 pseudowords were not able to be *proven* legal: 10 pseudowords (out of 100) created using 2-grams, 1 pseudoword (out of 100) created using 3-grams, and 1 pseudoword (out of 100) created using r-grams. Table 9 shows all of the CGCA pseudowords that contained a character sequence that did not exist in the origin wordlist. Four out of the five C+ errors contained a *y*, which should be considered a vowel in most cases, but was not treated as one for this evaluation. If *y* was treated as a vowel, these four pseudowords may not have incurred errors.

[TABLE 8 NEAR HERE] [TABLE 9 NEAR HERE]

The *individual suitability evaluation* was conducted manually by two of the researchers independently and then the results were compared. The results are shown in Table 10. For consistency, we agreed on using the list of affixes reported by Bauer and Nation to be either inflectional suffixes, or among the most frequently occurring and regular derivational affixes (Bauer and Nation 1993: 258-59), or frequent and orthographically regular affixes (Bauer and Nation 1993: 259-60), see Table 11.

[TABLE 10 NEAR HERE] [TABLE 11 NEAR HERE]

The manual coding was completed by two researchers separately. For the first three criteria (compound, polymorphic, and near polymorphic), any discrepancies were discussed and resolved. For the fourth criteria (one-character dissimilarity), only pseudowords marked positive by both researchers were included as positive in the final results. In coding the one-character dissimilarity, we did not use exhaustive dictionary searches but rather, we used our knowledge of English to see if any real word would immediately come to mind, without thinking too long (we will return to this later).

Discussion

Now, we turn to a discussion of the pseudowords that we generated using CGCA and their evaluation, provide a comparison of the same evaluation criteria applied to existing pseudoword generation systems currently in use, and discuss how these compare with the pseudowords generated by CGCA. Finally, we discuss cross-linguistic applicability of the CGCA system and conclude the section with some limitations and scope for future work.

Evaluation of the pseudowords generated by the CGCA algorithm

As regards legality of the pseudowords, we found that bi-grams performed the worst in our dataset, with 10% of the data containing non-legal pseudowords. This suggests caution in using systems that rely exclusively on bi-grams as means of generating pseudowords (i.e., WordGen). The next worst n-gram size was the 3-grams group, but this was comparatively better, with only 1% of the forms being non-legal. The system performed equally well for 3 or more n-gram sizes (the few non-legal elements in the r-gram set are likely to be bi-grams).

Turning to issues of suitability, we suggest that forms which only differ from existing words by one single character can be problematic as pseudowords because participants can mistakenly misread these forms as real words, particularly when under time pressure. This means that they might end up being unfairly penalised in a learner vocabulary test. This problem may seriously impact on systems that generate pseudowords by solely changing one single character (e.g., English Lexicon Project). But in coding the 1-char factor, we found that pseudowords which differ through a single letter from existing words are not all equally problematic: some are recognized much faster than others (recall that we coded this factor quickly, and without reference to a dictionary). For example, *overses* can easily be linked to the existing form *overseas*, whereas *orand* may not immediately be associated with *grand.* It is not straightforward to glean why some forms are immediately recognized while others are not. Words ending in "-in" and missing a "-g" might be recognized quickly due to the productivity of the progressive inflectional affix in English, but the picture is still incomplete. More work needs to be done in testing the forms on a larger population in order to understand the mechanism at work here.

Although compounding is a highly productive strategy for forming new words in English, we found the CGCA pseudowords generated using the BNC/COCA lists included strikingly few pseudo-compound forms, with a peak at 4-grams (10%). This suggests that 4-grams are the optimal character-gram size for generating compound words. It might be worthwhile to investigate experimentally whether there is a difference in how different pseudowords are viewed, based on structural differences (compound-like forms, versus polymorphic forms, versus morphologically simple forms).

In evaluating suitability, we have found interesting correlations between n-gram size and various factors. For *one-character dissimilarity*, there is a downward slope across n-grams, dropping from 2-grams (40%) to 4-grams (20%) before evening out. This trend suggests that the smaller the n-gram size, the more *easily* identifiable the pseudowords are to existing words (at 1 character away). Conversely, the occurrence of polymorphic pseudowords increases steadily from 2-grams (2%) to 8-grams (85%), suggesting that the larger the n-gram size, the more word-like (real root plus affix) the pseudowords are. Near-polymorphic pseudoword counts rise then fall, climbing from 2-grams (22%) up to 3-4-5-grams (40% each) and back down to 8-grams (9%). This final drop appears to be due to the corresponding climb in polymorphic occurrences. The combined counts of polymorphic and near-polymorphic pseudowords rise steadily until

they make up almost 100% (96% for 7-grams, 94% for 8-grams). These correlations may be due to how CGCA chains character-grams together to create pseudowords. The smaller the character-gram size, the fewer characters being compared and therefore the less likely that affixes will be generated. Likewise, the larger the character-gram size, the more characters being compared, and the more likely that affixes will be generated. Furthermore, pseudowords that include affixes may be less likely to be one character away from real words – they may instead be *n* characters away, where *n* is the length of the added affix. For example, *eightist* (polymorphic) is three characters (*ist*) away from the word *eight*, whereas *weat* (not polymorphic) is one character away from *wheat*.

Comparing CGCA pseudowords with pseudowords from existing Systems

We first used the main existing systems to generate 100 pseudowords from each (English Lexicon Project, the ARC Nonword Database, WordGen, Wuggy), and selected 100 pseudowords from Meara's EFL tests (20 from each level). The first 10 words from each are given for illustrative purposes in Table 12. We then conducted a *comparative legal evaluation*, where we used the legal evaluation criteria to evaluate both our pseudowords and each of the 100 pseudowords from the other systems. We decided to create a non-biased wordlist for this evaluation, rather than using our origin list or CELEX, drawing on four corpora: the Corpus of Historical American English (COHA) (Davies 2002), the Wikipedia Corpus (Davies 2015), News on the Web (NOW) (Davies 2013b), and Global Web-Based English (GloWbE)(Davies 2013a). All unique tokens were extracted from the corpora and only words that were validated as real words by Wiktionary were kept. The resulting wordlist contained 72,783 tokens.

[TABLE 12 NEAR HERE]

Each set of 100 pseudowords was compared against the COHA-Wikipedia-NOW-GloWbe wordlist and any character combinations (C+, V+, CV+C) that appeared in a pseudoword but not in the wordlist were noted (see Table 13). When comparing the CGCA pseudoword errors derived from the original wordlist with the CGCA pseudoword errors derived from the COHA-Wikipedia-NOW-GloWbe wordlist, the error counts for the CGCA pseudowords have: decreased from 10 to 9 for 2-grams, increased from 1 to 2 for 3-grams, increased from 0 to 1 for 4-grams, and stayed the same for all others (Table 13, first 9 rows). Comparatively, for the externally generated pseudowords: WordGen had the highest error count (26 out of 100 pseudowords contain errors), Wuggy had the second highest (21 out of 100), ARC and ELP had 16 and 10 respectively, and Meara's EFL pseudowords had the lowest error count, with only 6 out of 100 pseudowords containing errors. Surprisingly, ARC, WordGen, and Wuggy all included pseudowords that did not contain at least one vowel (*sprymphs, brft, grrpe, ymn*), although in the case of *sprymphs* (ARC), the *y* could be considered to be acting as a vowel.

[TABLE 13 NEAR HERE]

The next step was to conduct a *comparison suitability evaluation* with the same sets of pseudowords. Each pseudoword was manually coded following the same procedure as in the section titled "Designing post-production evaluation criteria and requirements" (Table 14). All have a relatively low count of *compound* pseudowords, however our 4-grams have the highest of them (10%), followed closely by Meara (9%). This suggests that using the CGCA algorithm with 4-grams should be preferred over other methods if aiming to generate compound pseudowords, but we stress here that suitability is highly dependent on the task. All of our *polymorphic* pseudowords, except those generated using 2-grams (2%), have higher counts than any of the externally generated pseudowords. Our highest come from our 7-grams (80%) and 8-grams (85%), while the highest in the externally generated pseudowords come from Meara (10%) and WordGen (8%), suggesting that the CGCA pseudowords are more word-like, in terms of a real stem plus affix. The *near-polymorphic pseudoword* counts are a little better balanced than for polymorphic pseudowords. ARC and ELP have the highest counts (58% each), followed by our 3-grams, 4-grams, and 5-grams (40% each). The lowest counts come from our 8-grams (9%), 7-grams (16%), and 6-grams (18%). Finally, for pseudowords that are easily identifiable as one character away from a real word (*character dissimilarity*), the highest counts come from WordGen (48%) and Wuggy (47%), followed closely by ELP (43%), 2-grams (43%), and 3-grams (39%). The lowest counts come from Meara (14%) and our r-grams (14%). Each of these statistics may be seen as either an advantage or a disadvantage, depending on the intended use of the pseudowords, and the form or structure required.

[TABLE 14 NEAR HERE]

The suitability considerations can be used to compare pseudowords created using different origins. For instance, if researchers wanted to create pseudowords using the CGCA, but that had the same structure as Meara's pseudowords, they could select those with similar compound, polymorphic, near-polymorphic, and character dissimilarity counts as his. Furthermore, as argued when discussing *Designing Post-Production Evaluation Criteria*, suitability criteria can be used to select more or less word-like pseudowords, for example polymorphic or near polymorphic pseudowords for

morphology experiments, and non-polymorphic pseudowords for vocabulary testing. The criteria can also be used to draw a comparison between sets of pseudowords, for example, if we wanted to create pseudowords that reflect the same form as some existing type (ELP, ARC, etc.).

Cross-linguistic application of the CGCA algorithm

The final set of observations concerns the linguistic domain of pseudowords. Here we ask whether pseudowords can be generated to reflect a particular language. Our solution was to develop the CGCA to work with any alphabet-based language, without requiring any knowledge of that language. The CGCA can be used to create pseudowords in any language because it just requires an origin wordlist. As an example, we have generated a sample of 10 pseudowords each from German, Spanish, and Italian using CGCA (using 4-grams), as shown in Table 15. The table also contains a sample of 10 English words. By specifying the desired language, we were able to use Wiktionary to validate each set of pseudowords. The three foreign language wordlists were derived from movie and television series subtitles from Buchmeier (2008a; 2008b; 2009). The German origin list was derived from the first 1,000 words in a frequency list of 25 million words (Buchmeier 2009); the Spanish origin list was derived from the first 1,000 words in a frequency list of 27 million words (Buchmeier 2008a); and the Italian origin list was derived from the first 1,000 words in a frequency list of 5.6 million words (Buchmeier 2008b).

[TABLE 15 NEAR HERE]

Moreover, given that all it requires is a wordlist or corpus, the legal evaluation criteria can be applied to pseudowords from any language, regardless of how they were created. The criteria can measure how well pseudowords fit within the legal orthographic form of any language, and are only limited by the size of the wordlist or corpus that is used. In conducting the legal evaluation on each of the language specific pseudowords, using their relative origin wordlists as the lexicon, we found that none of them violated any of the legal evaluation criteria.

CGCA can also be used to create domain specific or frequency specific pseudowords, it is limited only by the text that is used to build the origin wordlist. As an example, we have used CGCA (using 4-grams) to generate a sample of 10 pseudowords each from two different domains: Academic derived from the Academic Word List (Coxhead 2000) and Grade School, derived from the Basic Vocabulary Spelling List (Graham et al. 1993) (see Table 16).

[TABLE 16 NEAR HERE]

Limitations and future work

CGCA is only as good as its origin wordlist. If, for example, a general-purpose wordlist was used to generate pseudowords for a domain-specific vocabulary test, the resulting pseudowords would reflect the general-purpose language of the wordlist rather than the domain-specific language of the test. Similarly, the legal evaluation is only as good as the wordlist or origin that pseudowords are compared against. The results of the evaluation criteria would be affected by a wordlist that included misspelled words or partial words, for instance. We look forward to applications of CGCA in vocabulary and testing research, for example, using specialised wordlists such as Coxhead and Demecheleer (2018) in English for Specific Purposes, as well as wordlists in languages other than English, such as Jakobsen et al. (2018) in Danish.

Although CGCA does not require a large lexicon, the number of pseudowords that can be generated is proportionally related to the number of unique tokens in the origin wordlist. For example, when using an origin of 100 words, we were able to generate 100 pseudowords (using 3-grams), but with an origin of 1000 words we were able to generate 1000 pseudowords (using 3-grams). However, the larger the character-gram size, the fewer character-combinations there are, and therefore, fewer valid pseudowords can be generated. For example, when using an origin of 100 words, we were able to generate 100 pseudowords each using 2-grams, 3-grams, and r-grams, but only half as many using 4-grams.

CGCA uses Wiktionary to validate whether a potential pseudoword does or does not exist within the language. Although this is advantageous as it supports 8,000 languages, there are of course languages that it does not support fully (Te Reo Māori, for example). To address this problem, we intend to implement the algorithm as a web based solution, meaning researchers would be able to specify whether they wish to validate using Wiktionary, or use their own specially supplied wordlist for validation.

CGCA was implemented using the Python programming language and can be downloaded from GitHub.[iv] The bulk of the planned future work involves porting the Python code for CGCA over into an online web-based solution that can be made publicly available. The online version would allow researchers to upload an input corpus or wordlist, specify

whether they want the words in their corpus to be cleaned and validated, specify the size of the character-grams that they wish to use to create their pseudowords, specify the number of pseudowords they wish to generate, and specify whether they want to validate using Wiktionary or an uploaded wordlist. The system would then return the desired pseudowords for researchers to use as they wish.

Finally, majority of the pseudoword evaluations performed so far have focused on the English language. However, we are very interested in conducting more in-depth evaluations of CGCA pseudowords for other languages as well. Another aspect left for future research is comparing how participants in various tasks perform in relation to pseudowords obtained in different ways, and how their reaction times might vary.

Conclusion

This paper has introduced a new way of generating pseudowords that does not require any knowledge of the language and does not rely on a large lexicon. It uses a character-gram chaining approach to create pseudowords that reflect their origin, allowing us to create language or domain specific pseudowords with varying word-likeness.

We also argue that pseudowords need to be evaluated and propose two sets of criteria to this end – a legal evaluation and a suitability evaluation. The former evaluates character patterns against an origin to determine whether the pseudowords are *legal* within the language, while the later allows researchers to evaluate and compare the structure of pseudowords to determine *suitability*.

Supplementary material

SI consists of: a complete list of the 800 pseudowords generated by the CGCA for English, the legal scoring of both the CGCA pseudowords and the external pseudoword lists, and the manual suitability coding of the CGCA pseudowords and the external pseudoword lists. The CGCA algorithm can be downloaded from GitHub.[v]

Acknowledgements

All authors thank Emeritus Professor Ian Witten from the University of Waikato for forging collaborative networks between researchers at the University of Waikato and Victoria University of Wellington. JK thanks the University of Waikato Doctoral Scholarship for their financial support. AC thanks the Royal Society Marsden Fast Grant for their financial support.

Author contributions

All authors contributed a third of the work, and to all stages of the project.

Ethical statement

No data was collected from participants for this study, so no ethics was required. This project did not enlist the help of any participants, the authors have no conflict of interest and the experimental design was not shared with anyone else or required external validation.

References

Arndt, Henriette L and Woore, Robert (2018), 'Vocabulary learning from watching YouTube videos and reading blog posts', *Language Learning & Technology,* 22 (3), 124-42.

Baayen, R H, Piepenbrock, Richard, and van Rijn, Heddderick (1993), 'The CELEX database on cd-rom', (Web Download. Philadelphia: Linguistic Data Consortium).
Baayen, R Harald and Schreuder, Robert (2011), *Morphological structure in language processing* (151: Walter de Gruyter).
Balota, DA, et al. (2007), 'The English Lexicon Project', *Behaviour Research Methods*, 39 (3), 445-59.
Bauer, Laurie and Nation, I S P (1993), 'Word families', *International Journal of Lexicography*, 6 (4), 253-79.
Bell, Timothy C, Cleary, John G, and Witten, Ian H (1990), *Text compression* (348: Prentice Hall Englewood Cliffs, NJ).
Berko, Jean (1958), 'The child's learning of English morphology', *Word*, 14 (2-3), 150-77.
Buchmeier, M (2008a), *Bilingual Dictionaries for Offline Use - Spanish frequency list*.
--- (2008b), *Bilingual Dictionaries for Offline Use - Italian frequency list*.
--- (2009), *Bilingual Dictionaries for Offline Use - German frequency list*.
Cardenas, Jessica M (2009), 'Phonics instruction using pseudowords for success in phonetic decoding', (Florida International University).
Coxhead, Averil (2000), 'A new academic word list', *TESOL quarterly*, 34 (2), 213-38.
Coxhead, Averil and Demecheleer, Murielle (2018), 'Investigating the technical vocabulary of Plumbing', *English for Specific Purposes*, 51, 84-97.
Davies, Mark (2002), *The Corpus of Historical American English (COHA)*.
--- (2013a), *Global Web-Based English (GloWbE)*.
--- (2013b), *News on the Web (NOW)*.
--- (2015), *The Wikipedia Corpus*.
Duyck, Wouter, et al. (2004), 'WordGen: A tool for word selection and nonword generation in Dutch, English, German, and French', *Behavior Research Methods, Instruments and Computers*, 36 (3), 488-99.
Elgort, Irina and Warren, Paul (2014), 'L2 vocabulary learning from reading: Explicit and tacit lexical knowledge and the role of learner and item variables', *Language Learning*, 64 (2), 365-414.
Graham, Steve, Harris, Karen R, and Loynachan, Connie (1993), 'The Basic Spelling Vocabulary List', *The Journal of Educational Research*, 86 (6), 363-68.
Groff, P 'The Usefulness of Pseudowords', < http://www.nrrf.org/old/essay_pseudowords.html>, accessed.
Jakobsen, Anne Sofie, Coxhead, Averil, and Henriksen, Birgit (2018), 'General and academic high frequency vocabulary in Danish', *Nordand*, 2 (01), 64-89.
Keuleers, Emmanuel and Brysbaert, Marc (2010), 'Wuggy: A multilingual pseudoword generator', *Behavior research methods*, 42 (3), 627-33.
Kučera, Henry and Francis, Winthrop Nelson (1967), *Computational analysis of present-day American English* (1 edn.: Brown University Press).
Meara, Paul (2010), *EFL Vocabulary Tests* (second edn.; Swansea: Lognostics.).
Miller, George Armitage and Selfridge, Jennifer A (1950), 'Verbal context and the recall of meaningful material', *The American Journal of Psychology*, 63, 176-85.
Nation, I.S.P, Heatley, A, and Coxhead, Averil (2002), 'Range: A program for the analysis of vocabulary in texts [software]'.
New, Boris, et al. (2004), 'Lexique 2: A new French lexical database', *Behavior Research Methods, Instruments and Computers*, 36 (3), 516-24.
Nordquist, R 'Definition and Examples of Pseudowords', <https://www.thoughtco.com/pseudoword-definition-1691549>, accessed.
Proença, Jorge, et al. (2017), 'Automatic evaluation of children reading aloud on sentences and pseudowords', *Proc. INTERSPEECH*, 2749-53.
Rastle, Kathleen, Harrington, Jonathan, and Coltheart, Max (2002), 'The ARC nonword database', *The Quarterly Journal of Experimental Psychology*, 55 (4), 1339-62.
Rueckl, Jay G and Olds, Elizabeth M (1993), 'When pseudowords acquire meaning: Effect of semantic associations on pseudoword repetition priming', *Journal of Experimental Psychology: Learning, Memory, Cognition*, 19 (3), 515.
Saragi, T, Nation, I S P, and Meister, G F (1978), 'Vocabulary learning and reading', *System* 6(2), 72-78.
Schwartz, Steven (2013), *Measuring reading competence: A theoretical-prescriptive approach* (Springer Science & Business Media).
Webb, Stuart (2005), 'Receptive and productive vocabulary learning: The effects of reading and writing on word knowledge', *Studies in Second Language Acquisition*, 27 (1), 33-52.
--- (2007), 'The effects of repetition on vocabulary knowledge', *Applied Linguistics*, 28 (1), 46-65.

Figures

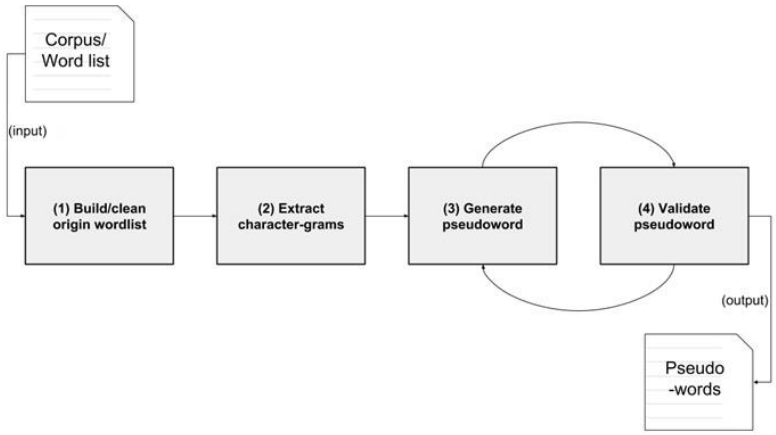

Figure 1: The modular steps involved in the character-gram chaining algorithm

Tables

Table 1: Creating pseudowords by manipulating a stimulus

Insertion (h)	Deletion (o)	Transposition (i)	Composition (ation)
piloth	pilt	pliot	pilotation

Table 2: Creating a pseudoword (*reroin*) using high frequency bi-grams. Generated using WordGen (Duyck et al. 2004)

bigram:	re	er	ro	Oi	in
frequency:	4760	7279	2840	468	7156

Table 3: Creating a pseudoword (*shib*) by combining sub-syllabic elements

Onset	Nucleus	Coda	Pseudoword
sh (as in *show*)	i (as in *tin*)	b (as in *bib*)	Sh-i-b

Table 4: Natural language modelling with n-gram models (Bell et al. 1990)

Order-0 text (single character)	*fsn'iaad ir lntns hynci,..aais oayimh t n ,at oeotc fheotyi t afrtgt oidtsO, wrr thraeoe rdaFr ce.g*
Order-5 text (6-gram)	*Number diness, and it also light of still try and amoung Presidential discussion is department-trans*
Order-11 text (12-gram model)	*Papal pronounced to the appeal, said that he'd left the lighter fluid, ha, ha"? asked the same numbe*

Table 5 : A sample of 10 pseudowords generated by CGCA (prior to evaluation)

2-gram	3-gram	5-gram	8-gram	r-gram
scon	punit	untalentleman	uncertification	eightist
cens	recollusted	unlabelling	representably	braveller
nes	cree	registract	unstructure	unexception
vois	dward	injusting	undifference	disbehaviour
sunt	witle	orches	intergovernment	ninthood
zer	captime	heritancy	unconsolidate	apartmentalizing
stro	hydraft	easters	uncirculates	lettes
ghol	natigating	unsenting	semanticise	gotters

acive	ouncing	essentee	undistinguish	greeness
weat	runnius	impatibly	reaffirmative	whitecturalisation

Table 6: Criteria for assessing pseudoword legality (for English)

C+ (one or more consecutive consonants)	Extract sequences of consecutive consonants from a pseudoword and validate them only if they appear within a token in the origin wordlist. For example, for the pseudoword *conferious,* its consecutive consonants are: *c*, *nf*, *r*, and *s*.
V+ (one or more consecutive vowels)	Extract sequences of consecutive vowels from a pseudoword and validate them only if they appear within a token in the origin wordlist. For example, for the pseudoword *conferious,* its consecutive vowels are: *o*, *e*, and *iou*.
CV+C (a consonant, followed by one or more consecutive vowels, followed by a consonant)	Extract sequences of consecutive vowels including one leading and one trailing consonant from a pseudoword and validate them only if they appear within a token in the origin wordlist. For example, for the pseudoword *conferious,* its cv+c patterns are: *con*, *fer*, and *rious*.

Table 7: Criteria for assessing pseudoword suitability (for English)

Compound	A pseudowords that is made up of two or more real words within the language. For example captime (cap-time).
Polymorphic	A pseudoword that consists of a real root plus one of more affixes. For example in-determines (in-determine-s). Note that a compound can also be polymorphic. For example captimed (cap-time-ed).
Near polymorphic	A pseudoword whose root does not exist in the language but includes one or more affixes. For example alphise (alph-ise). Note that a pseudoword can be either polymorphic or near polymorphic, but not both. It either has a real root or it doesn't.
One-character dissimilarity	A pseudoword that is easily identifiable as one character away from a real word within the language. For example overses (overseas).

Table 8: Results for the individual legal evaluation (per 100 pseudowords)

Pseudowords	C+ errors	V+ errors	CV+C errors	Non-legal words
2grams	4	0	6	10
3grams	0	0	1	1
4grams	0	0	0	0
5grams	0	0	0	0
6grams	0	0	0	0
7grams	0	0	0	0
8grams	0	0	0	0
r-grams	1	0	0	1

Table 9: Error examples for the individual legal evaluation

Category	Pseudoword	C+ Errors	V+ Errors	CV+C Errors
2gram	yies	0	0	1
2gram	yied	0	0	1
2gram	vois	0	0	1
2gram	gymma	1	0	0
2gram	tbscrap	1	0	0
2gram	faugh	0	0	1
2gram	eiguit	0	0	1
2gram	jous	0	0	1
2gram	dyntin	1	0	0
2gram	gympart	1	0	0
3gram	reuniour	0	0	1

r-gram	rhydrate	1	0	0

Table 10: Results from the suitability evaluation (per 100 pseudowords)

Category	Compound	Polymorphic	Near Polymorphic	Char Dissimilarity
2-gram	3	2	22	43
3-gram	5	10	40	39
4-gram	10	22	40	20
5-gram	5	44	40	18
6-gram	1	71	18	16
7-gram	1	80	16	21
8-gram	4	85	9	20
r-gram	5	47	33	14

Table 11: Chosen coded affixes, derived from levels 2, 3, and 4 by Bauer & Nation (1993)

Inflectional suffixes	Frequent and regular derivational affixes	Frequent orthographically regular affixes
-s, -ies, -es, -ed / -d / -t, -en, -ing, -er, -es	-able, -er, -ish, -less, -ly, -ness, -th, -y, non-, un-	-al, -ation, -ess, -ful, -ism, -ist, -ity, -ize / -ise, -ment, -ous, in- / im-

Table 12: A sample of 10 pseudowords generated using ARC, ELP, WordGen, Wuggy, and Meara

ARC	ELP	WordGen	Wuggy	Meara
grev	drimaced	daney	dre	berrow
bloap	nightkine	biled	woubt	whaley
shrusks	sonehead	ragio	istye	contrivial
zoc	creemason	applk	hu	detailoring
spails	selectove	hoory	roud	eldred
gir	nonclude	loeer	pliedes	gumm
thwiped	gastrami	adoke	onsce	pocock
grear	asjoins	cheed	buit	pernicate
prirr	guinbess	flort	fims	eluctant
crenched	egocative	fraze	sussest	limidate

Table 13: Results from the Comparison Legal Evaluation (per 100 pseudowords)

Pseudowords	C+ errors	V+ errors	CV+C errors	Non-legal words
2grams	4	0	5	9
3grams	0	0	2	2
4grams	0	0	1	1
5grams	0	0	0	0
5grams	0	0	0	0
6grams	0	0	0	0
7grams	0	0	0	0
8grams	0	0	0	0
r-grams	1	0	0	1
ARC	3	1	14	16
ELP	4	2	6	10
WordGen	8	5	18	26

Wuggy	3	1	19	21
Meara	0	0	6	6

Table 14: Results from the Comparison Suitability Evaluation (per 100 pseudowords)

Category	Compound	Polymorphic	Near Polymorphic	Char Dissimilarity
2-gram	3	2	22	43
3-gram	5	10	40	39
4-gram	10	22	40	20
5-gram	5	44	40	18
6-gram	1	71	18	16
7-gram	1	80	16	21
8-gram	4	85	9	20
r-gram	5	47	33	14
ARC	1	1	58	34
ELP	6	3	58	43
WordGen	1	8	36	48
Wuggy	3	7	37	47
Meara	9	10	26	14

Table 15: Language specific pseudowords generated using the CGCA

German	Spanish	Italian	English
bisscheint	mirande	abbastardo	acknowier
kinden	puestra	dicevuto	reorganic
viellen	suficio	dentre	sweaten
wassen	oporta	momente	clinist
alleich	histos	dimente	inflatting
scheinlich	tambiar	ufficile	puddiness
entschuld	suerto	pagari	tonnect
viellein	grando	ottimana	incling
entschule	accidentro	finalmeno	epidest
bisscheiße	dentra	lavore	prograph

Table 16: Domain specific pseudowords generated using the CGCA

Academic	Grade school
unconverse	brough
enormat	brothes
corresponse	withough
illustract	mountries
emergins	grandmothes
primarise	countain
majoritise	cottom
phasize	mountry
undiminution	clother
preliminish	botton

□□

[i] One anonymous referee points out that pseudowords generated by manipulating characters within words can be problematic for some languages, particularly those with more rigid (and easily recognizable) syllable structure, such as, Italian, Spanish, and Arabic. While the algorithm we introduce here does not specifically take syllabic structure into consideration, the fact that it pays close attention to letter combinations is likely to lead to a close resemblance between syllabic structure of real words in the language and the pseudowords generated. Ultimately, we feel that this further highlights the importance of evaluating any pseudowords generated, regardless of the method used to do so.

[ii] We use the term *origin* to refer to the list of unique words that are extracted from the input text.

[iii] We thank one of the anonymous reviewers for reminding us of this fitting example.

[iv] https://github.com/jlkonig/

[v] https://github.com/jlkonig/