Generating Nonwords for Vocabulary Proficiency Testing

Osama Hamed, Torsten Zesch

Language Technology Lab
Department of Computer Science and Applied Cognitive Science
University of Duisburg-Essen, Germany
{osama.hamed, torsten.zesch}@uni-due.de

Abstract

Lexical recognition tests are frequently used for measuring language proficiency. In such tests, learners need to differentiate between words and artificial nonwords that look much like real words. Our goal is to automatically generate word-like nonwords which enables repeated automated testing. We compare different ranking strategy and find that our best strategy (a specialized higher-order character-based language model) creates word-like nonwords. We evaluate our nonwords in a user study and find that our automatically generated test yields scores that are highly correlated with a well-established lexical recognition test which was manually created.

1. Introduction

Lexical recognition tests (Meara and Jones, 1987) are frequently used for measuring language proficiency. In such a test, students are typically shown either valid words (*denial*) or nonwords (*platery*), and need to decide if they are valid or not. The nonwords used in lexical recognition tests should look like words without being actually in the lexicon. Thus, *platery*, *interfate*, or *purrage* have been shown to work well while *abcde* or *autobahn* are less suitable. The main advantage of lexical recognition testing is its simplicity. It only takes five minutes, only "Yes/No" questions are asked, and scoring can easily be automated.

The task of having students recognize words for vocabulary proficiency testing goes back quite a long time, cf. (Schmitt, 2000). The Eurocentres Vocabulary Size Test (Meara and Jones, 1987) is an early example of using nonwords for testing. They used 150 items – two thirds real words and one third nonwords. Lemhöfer and Broersma (2012) create an adapted version called LexTALE that can be finished faster, as it only uses 60 items. They validate the resulting scores by correlating them with other proficiency scores based on a word translation task and the commercial 'Quick Placement Test'.

LexTALE has been adapted to other languages beyond English, e.g. Dutch and German (Lemhöfer and Broersma, 2012), French (Brysbaert, 2013), or Spanish (Izura et al., 2014). Using nonwords constitutes an improvement over other forms of vocabulary proficiency testing, as it simplified the setup. For example, the Vocabulary Levels Test (Nation, 1990) is based on matching words with definitions, which is much harder to administer and automate.

In the past, nonwords have been manually created, but for repeated testing as used in formative assessment (Wang, 2007) we need to be able to generate them automatically. Thus, in this paper we explore methods for automatically generating good nonwords.

2. Generating Nonwords

We model the selection of word-like nonwords as a two-step process where we first generate candidate strings and then rank them according to their 'wordness'.

2.1. Candidate Selection

We generate random strings of different length and check against a list of known English words in order to ensure that we only have nonword candidates. This strategy will obviously create a lot of bad nonwords, which have little resemblance with known words. However, more informed strategies might already use the same information as will be later used for ranking and thus bias the results.

2.2. Candidate Ranking

In this section, we describe the different ranking strategies used to find good (i.e. word-like) nonwords.

Random Baseline This is a simple baseline that randomly orders the nonwords. It is mainly used to set the other results into perspective.

Neighbourhood Size (nh-size) We compute the edit distance between a generated nonword and all words from a dictionary with known English words. We then rank the candidates according to the number of English words with low edit distance ($k = 1$ in our case). This means that nonwords having more orthographic neighbors are being ranked higher, which is a simple approximation for the probability that a learner confuses a nonword with a known word from the lexicon (Duyck et al., 2004).

Character Language Model This set of ranking methods is motivated by the observation that words in a language contain certain characteristic character combinations that make them look like a valid word of that language. For example, the word *großzügig* might look vaguely German to you even if you don't speak German.[1] This fact is also used in language identification where character language models are frequently used in order to distinguish languages (Cavnar et al., 1994; Vatanen et al., 2010). We are going to use character language models with the goal to find nonwords like *platery* that look English, but actually are not part of the lexicon. We experiment with unigram, bigram, and trigram models, but expect higher-order language models to work better. For ranking, we as-

[1] It means *generous* in English.

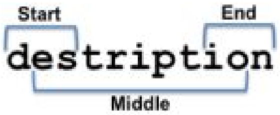

Figure 1: Example for position specific splitting. Each part is scored with its own character language model.

Pos	Ranking #1	P	Ranking #2	P
1	LT	1.00	nonword	-
2	nonword	-	LT	0.50
3	LT	0.67	nonword	-
4	LT	0.75	nonword	-
5	nonword	-	LT	0.40
6	nonword	-	nonword	-
7	nonword	-	nonword	-
8	nonword	-	nonword	-
9	nonword	-	nonword	-
10	nonword	-	LT	0.30
	AP	**0.81**		**0.40**

Table 1: Example for computing average precision (AP) for two different rankings. Whenever an LexTALE (LT) word is observed, precision P is computed for this subset.

sign to each word its probability returned by the language model.

Position Specific A drawback of the simple character language model is that it assigns equal probability to a character n-gram no matter where it appears in a word. However, it is clear that the trigram *ing* is more likely at the end of a word than at the beginning. We thus augment the simple model to include position specific information following Duyck et al. (2004).

As the importance of the first and last letters of each word for reading is well known (Johnson and Eisler, 2012), we break each string into three parts: *start*, *middle*, and *end*. Figure 1 shows an example of our split. For each part, we separately train and apply a position-specific character language model.

3. Experimental Setup

In our experiments, we want to find the best ranking strategy, where we expect higher-order n-gram models to work better, and position specific language models to outperform corresponding simple models. We train all language models using the Brown Corpus (Francis and Kuçera, 1964). We deliberately used a rather small corpus to show that character language models do not need much training data.

3.1. Evaluation Metric

In order to measure the quality of a ranking, we need to know whether word-like nonwords are ranked on the top positions. For that purpose, we are taking the 21 nonwords from LexTALE lexical recognition test (Lemhöfer and Broersma, 2012) as a gold standard. They are known to be easily confused with real words, which means that a good ranking function should rank them at the top.

	nonword length (characters)								
Strategy	**All**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
random	.00	.00	.00	.00	.00	.00	.00	.00	.00
nh-size	.02	.00	.07	.29	.53	.57	.57	.57	.57
1-gram	.02	.01	.01	.02	.03	.05	.09	.11	.14
1-gram-PS	.06	.04	.04	.04	.06	.07	.09	.11	.14
2-gram	.13	.04	.07	.14	.26	.41	.54	.64	.72
2-gram-PS	.43	.19	.26	.40	.53	.65	.73	.79	.83
3-gram	.30	.09	.19	.38	.54	.69	.78	.85	.90
3-gram-PS	.67	.41	.55	.67	.75	.81	.84	.87	.89

Table 2: Average precision of ranking strategies

LexTALE	**Ranked nonwords**	
nonwords	top-10	bottom-10
platery	ahers	zlkcltmirk
destription	dand	ydbwehwve
alberation	whil	oumacivcgi
mensible	lign	dkucrxuvhvi
interfate	folli	lzurtqsrv
proom	golay	athfiprzbjq
fellick	poteru	qocbuabvh
exprate	alopirdrel	vnesfrqqjt
rebondicate	hindscomy	bgicpzycl
purrage	sherotspia	kcnkqpgt

Table 3: The top-10 LexTALE nonwords (LTs); top-10 and bottom-10 nonwords as per the ranking of 10K randomly generated nonwords using 3-gram-PS approach.

As evaluation metric, we are utilizing average precision (AP) from information retrieval. Table 1 gives an example showing two example rankings. Each time we find one of our gold standard nonwords from the LexTALE (LT) list, we compute the precision at that point taking only into account the items retrieved so far. For example in ranking #1, we find an LT nonword at the first position. As all items retrieved so far are LT nonwords, the precision is 1. The next LT nonword is on position 3. At this point, we have retrieved 2 LT items and 1 candidate nonword item which results in a precision of $^{2}/_{3}$. The third LT nonword in ranking #1 is found on position 4, for a precision of $^{3}/_{4}$. Average precision is now computed as the average over the three precision values. Computing the average precision in the same way for ranking #2 confirms that #1 is much better than #2.

3.2. Evaluation Dataset

In order to create the evaluation dataset, we generate 10,000 random nonwords with length between 4 and 11 letters (the same length limits as in LexTALE). We then add the 21 gold standard nonwords from LexTALE that are going to be used for evaluation. In order to smooth the results, we repeat the experiment 100 times (generating new random nonwords every time) and report mean average precision values.

4. Results

Table 2 shows the average precision values for all nonwords in the dataset as well as per nonword length. From the table, we can see that the random baseline is close to zero showing that our dataset size of 10,000 candidates is large enough to avoid random strategies to have any effect. Neighborhood size does not work well in general, which is especially due to the bad performance on the shorter nonwords, while it works reasonably well on the longer ones. For the language model based approaches, we observe two trends which are in line with our hypotheses: (i) higher n-gram models work better, and (ii) position specific models always work better than the simple model. Our best strategy is thus the 3-gram position specific ranking with an average precision of 0.67, which means that almost all gold standard nonwords are ranked very high among the 10,000 candidates. The breakdown of results per nonword length shows that longer nonwords are generally easier to rank which can be explained by the fact that the score of longer nonwords is more difficult to influence by a single very frequent n-gram.

In Table 3, we show some examples of the LexTALE nonwords that we use as a gold standard. We also show the top-10 as well as the bottom-10 candidates as ranked by our best strategy. The top-10 looks much more work-like compared to the bottom-10 showing that our ranking is effective, but compared with the gold standard LexTALE words, our generated nonwords seem to be of lower quality. However, this is only an informal evaluation and it is unclear whether the perceived difference will have any effect in an actual lexical recognition test. Thus, in the next section we formally compare our test with LexTALE in a user study.

5. User Study

The goal of the user study is to test how well our generated nonwords work in a lexical recognition test compared to an established test like LexTALE.

5.1. Selecting Words

For our test, we use the nonwords generated by our best strategy (3-gram-PS) as described above. However, besides nonwords, we also need a suitable set of known English words. Ideally, they should span the whole difficulty range from simple to sophisticated. We follow Lemhöfer and Broersma (2012) who select words from different ranges of relative frequency in a large corpus. This makes use of the well established fact that there is a high correlation between the frequency of a word and its difficulty (Greenberg, 1965). According to Duyck et al. (2004), this also ensures a better comparability when the test is conducted for different languages.

We use the Brown corpus (Francis and Kuçera, 1964) in order to determine the relative frequency of words. We follow the LexTALE procedure and randomly select words with 4 to 12 letters[2] and a corpus frequency between between 1 and 26 occurrences per million words. We also make sure to select the same number of words from different word classes as in LexTALE. However, many English words have multiple word classes, so an exact mapping from out-of-context words into word classes is not possible anyway. The resulting list of words is shown in Table 4.

Class	Set
Nouns (15)	canto, hilt, quantum, leeway, barbell, vintage, allegory, fable, pallor, shovel, tavern, huddle, primacy, gadfly, syndicate
Adjectives (12)	intermittent, turbulent, appreciative, parasitic, snobbish, arrogant, lusty, exquisite, endurable, reverent, orchestral, septic
Adverbs (2)	lengthwise, precariously
Verbs (11)	mold, forfeit, veer, enrich, rape, intervene, expel, strut, buckle, blend, forestall

Table 4: Set of words used in our test categorized by word classes.

5.2. Setup

We asked participants to complete a three-part study: (i) a self-assessment of English language proficiency, (ii) the manually created LexTALE test, and (iii) our automatically generated test. We utilize Moodle[3] (a well-known learning management system) to conduct the study.

First, we provide participants with a set of instructions including some sample items. Then the participants were asked to provide information about gender, age, L1, the number of years they had taken English courses in school, and the self-rated language proficiency using Common European Framework of Reference (CEF)[4] levels. Finally, participants had to finish the LexTALE test and our test. In order to avoid sequence effects, participants randomly either get LexTALE first and then our test, or vice versa. However, we do not randomize the order of items within a test following the LexTALE guidelines.

Scoring For each participant, we compute the test score for both tests using the scoring scheme introduced by (Lemhöfer and Broersma, 2012). In order to account for the unequal number of words and nonwords in the test, it averages the corresponding accuracies:

$$score = \frac{(a_w + a_{nw}) \cdot 100}{2} \qquad (1)$$

where a_w is the accuracy on words and a_{nw} on nonwords.

5.3. Study Results

We recruited 80 participants from two German universities, but only 45 finished all three parts of the study. 23 are female, 28 are German native speakers, and the average age is 22.4 years.

In order to compare the quality of our test with the original LexTALE test, we compute for each student the test

[2] This is a different size compared to nonwords in LexTALE that are 4 to 11 letters long. In order to ensure comparability with LexTALE, we follow those length constraints, but newly generated tests should use the same constraints for words and nonwords.

[3] https://moodle.org

[4] http://www.englishprofile.org/index.php/the-cef

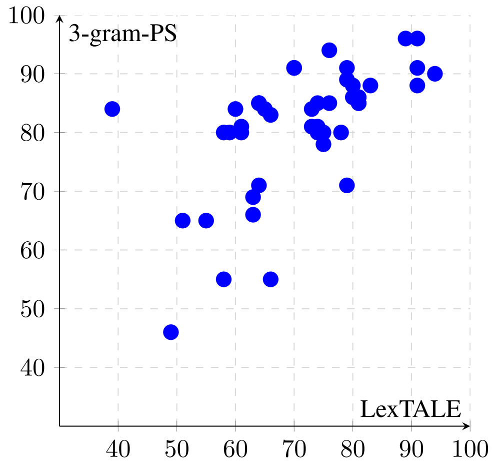

Figure 2: Participants' scores on original LexTALE test vs. the test generated by our approach. Original scoring function.

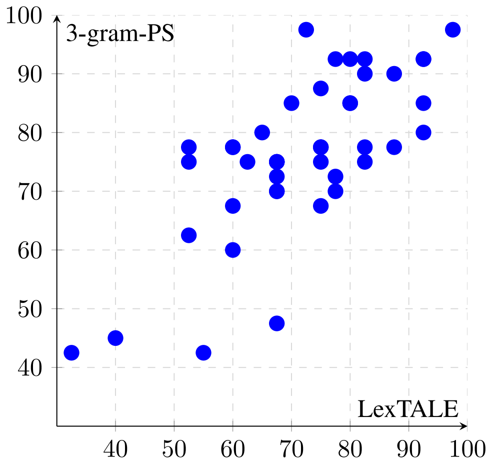

Figure 3: Participants' scores on original LexTALE test vs. the test generated by our approach. Adapted scoring function using only words.

score according to formula (1) and then compute Spearman correlation ρ between the resulting score vectors for both tests. We obtain a correlation of 0.68 and Figure 2 shows the corresponding scatterplot. We see that our test assigns vocabulary proficiency scores close to the ones assigned by LexTALE, but that there are some outliers.

In order to further analyze the differences between the two tests, we show a breakdown of accuracy for correctly detecting words vs. correctly rejecting nonwords in Table 5. We see that the accuracy for words is almost the same for both tests (.70 vs. .73), while our nonwords are much easier to recognize (.90 accuracy) compared to LexTALE (.75). This indicates that our nonwords do have lower quality compared to the LexTALE nonwords, as we suspected in Section 4. Interestingly this has little effect on the words, i.e. they do not get easier even if the nonwords are easier. This is probably due to the fact that nonwords do only need to be of reasonable quality in order to force students to make mistakes on the words. As a consequence, including the nonwords into the scoring might not be necessary at all, as we are only interested in how well participants recognize words, while the nonwords are only distractors.

If we drop the nonword part from equation (1), we can directly use the accuracy on words a_w as the test score. We obtain a correlation is 0.70 (compared to 0.68 from above when taking also nonwords into account). Figure 3 shows the corresponding scatterplot. Even if the correlation only improves slightly, the score distribution is much better with fewer outliers.

We can conclude that in the light of the high correlation between the two tests, our automatically generated test is as effective as the manually created LexTALE in measuring the vocabulary proficiency level of learners.

Self ratings Lemhöfer and Broersma (2012) provide a partial mapping from the test score to CEF levels, where scores equal to 59 points and below are mapped to B1 (or lower), scores between 60 and 79 points are mapped to B2, and scores above 80 points are mapped to C1 (or higher). The mapping is only partial because the test is not able to distinguish well for very early and very advanced stages of learning. We map the LexTALE scores and our scores to CEF levels using the a_w score. In 73% of all cases, both test agree on the same level (with a 33% chance of random agreement).

LexTALE		3-gram-PS	
a_w	a_{nw}	a_w	a_{nw}
.70	.75	.73	.90

Table 5: Accuracy of student responses for words a_w and nonwords a_{nw}.

CEF levels also allow us to compare with the self ratings, but this needs to be taken with a grain of salt, as we have no way of knowing how accurate the self ratings actually are. LexTALE assigns the self rated level in 40% of all cases, and our test in 49% showing again that both tests behave quite similar.

6. Related Work

The two main paradigms for creating nonwords (either manually or automatically) are (i) to start from a known word and change it to get a nonword, or (ii) to use smaller units (letters, syllables) to construct a larger nonword string.

The first paradigm was followed by the English Lexicon Project[5] (Balota et al., 2007), where they constructed a nonword database by manually changing one or more letters starting with known English words.

An example for the second paradigm is the ARC nonword database (Rastle et al., 2002) that contains monosyllabic nonwords which follow the phonotactic and orthographic rules of (Australian) English. The database only provides the nonwords, but does not rank them according to their quality. Another approach is WordGen (Duyck et al., 2004), which is an interactive tool for generating

[5] http://elexicon.wustl.edu

nonwords. It supports both paradigms and lets the user manipulate nonword properties that are similar to the ones we use for ranking, e.g. neighborhood size, position specific bigram frequency etc. In the end, the user is supposed to pick suitable nonwords, while our approach is fully automatic. Wuggy (Keuleers and Brysbaert, 2010) builds on WordGen but introduces syllable template to build nonwords that more closely resemble a certain word.

All those approaches are more geared towards psycholinguistic research letting researchers select suitable nonwords or generate nonwords that are similar to a given word. In contrast our approach is supposed to work fully automatic and to create a new list of high quality nonwords whenever a lexical recognition test needs to be conducted.

7. Conclusion

We have tackled the task of automatically generating nonwords for lexical recognition tests. We show that character language models can be used to distinguish low and high quality nonwords, and that higher-order models incorporating position specific information work best. We evaluate the generated nonwords in a user study showing that our approach yields test scores that are highly correlated with the scores obtained from an established lexical recognition test. The study also shows that the difficulty of the nonwords has little effect on how well words are recognized. Nonwords only act as distractors forcing students to make mistakes on the words. Thus, scoring can be simplified to ignore nonword performance which even slightly increases the correlation between both tests. With our experiments, we have shown that lexical recognition tests for English can be fully automatically created.

In future work, we want to verify our findings for other languages, and try approaches that generate and rank at the same time like Markov models (Westbury et al., 2007).

References

Balota, David A, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman, 2007. The English Lexicon Project. *Behavior Research Methods*, 39(3):445–459.

Brysbaert, Marc, 2013. LexTALE_FR a fast, free, and efficient test to measure language proficiency in French. *Psychologica Belgica*, 53(1):23–37.

Cavnar, William B, John M Trenkle, et al., 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175.

Duyck, Wouter, Timothy Desmet, Lieven PC Verbeke, and Marc Brysbaert, 2004. Wordgen: A tool for word selection and nonword generation in dutch, english, german, and french. *Behavior Research Methods, Instruments, & Computers*, 36(3):488–499.

Francis, W. Nelson and Henry Kuçera, 1964. Manual of Information to Accompany a Standard Corpus of Present-day Edited American English, for use with Digital Computers.

Greenberg, Joseph H, 1965. Some generalizations concerning initial and final consonant sequences. *Linguistics*, 3(18):5–34.

Izura, Cristina, Fernando Cuetos, and Marc Brysbaert, 2014. Lextale-esp: A test to rapidly and efficiently assess the spanish vocabulary size. *Psicologica: International Journal of Methodology and Experimental Psychology*, 35(1):49–66.

Johnson, Rebecca L and Morgan E Eisler, 2012. The importance of the first and last letter in words during sentence reading. *Acta psychologica*, 141(3):336–351.

Keuleers, Emmanuel and Marc Brysbaert, 2010. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633.

Lemhöfer, Kristin and Mirjam Broersma, 2012. Introducing lextale: A quick and valid lexical test for advanced learners of english. *Behavior Research Methods*, 44(2):325–343.

Meara, Paul and Glyn Jones, 1987. Tests of vocabulary size in english as a foreign language. *Polyglot*, 8(1):1–40.

Nation, Paul, 1990. Teaching and learning vocabulary. Rowley, MA: Newbury House.

Rastle, Kathleen, Jonathan Harrington, and Max Coltheart, 2002. 358,534 nonwords: The arc nonword database. *The Quarterly Journal of Experimental Psychology: Section A*, 55(4):1339–1362.

Schmitt, Norbert, 2000. *Vocabulary in language teaching*. Ernst Klett Sprachen.

Vatanen, Tommi, Jaakko J Väyrynen, and Sami Virpioja, 2010. Language identification of short text segments with n-gram models. In *LREC*. Citeseer.

Wang, Tzu-Hua, 2007. What strategies are effective for formative assessment in an e-learning environment? *Journal of Computer Assisted Learning*, 23(3):171–186.

Westbury, Chris, Geoff Hollis, and Cyrus Shaoul, 2007. Lingua: the language-independent neighbourhood generator of the university of alberta. *The Mental Lexicon*, 2(2):271–284.

DOI: 10.17185/duepublico/72102
URN: urn:nbn:de:hbz:464-20211022-103534-2

Hamed, Osama/Zesch, Torsten (2015): Generating Nonwords for Vocabulary Proficiency Testing. In: *Proceedings of the 7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. Poznań, Poland, 2015*, pp. 473–477. http://ltc.amu.edu.pl/a2015/book/papers/APP-1.pdf