ALRT: Cutting Edge Tool for Automatic Generation of Arabic Lexical Recognition Tests

Osama Hamed
Computer Systems Engineering Department
Palestine Technical University
Tulkarm - Palestine
osama.hamed@ptuk.edu.ps

Saeed Salah
Department of Computer Science
Al-Quds University
Abu Dis, Jerusalem, 20002, Palestine
sasalah@staff.alquds.edu

Abed Alhakim Freihat
Department of Computer Science
University of Trento
Trento - Italy
abed.freihat@unitn.it

Abstract

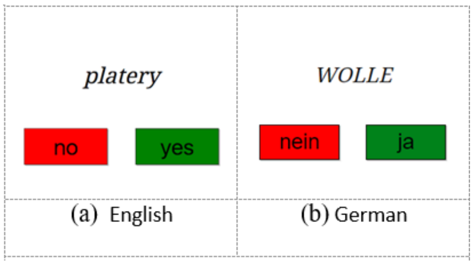
A Lexical Recognition Tests (LRT) is a common tool being widely used to measure the level of language-learner's proficiency utilizing vocabulary size (or simply the number of words acquired by a learner) for several international languages like English, Arabic, German, Chinese, and Spanish. Compared to other languages, LRT themes for Arabic are not mature enough and still they have some rooms for improvement, with very few existing proposals that mainly use human-crafted or semi-automated methods using Arabic Natural Language Processing (NLP) techniques. This paper introduces ALRT, the Arabic Lexical Recognition Tests Tool for the automatic generation of Arabic LRTs. The tool was tested using a huge dataset of Arabic vocabulary, and a subject-matter expert intervention was involved as an extra validation step to verify the quality of generated nonwords.

1 Introduction

Arabic is one of the main languages being widely used. It is not only spoken by more than 422 million people, but also non Arab people are using Arabic to practice Islam, study Arab cultures, and collect Arabs' opinions about many topics, etc. (Abdelgadir and Ramana, 2017). Arabic is mainly divided into three classes; standard, spoken and classical (Elfardy and Diab, 2012). The standard Arabic is the language used for official documents, language learning centers, and educational resources and books; the spoken Arabic constitutes the main spoken language of Arabs in modern society, it has many dialects that represent various diversities of the real spoken Arabic language - Levant, Moroccan, Gulf, Levant, and Egyptian- which leads to to so-called Arabic "Diglossia", i.e., Arab people use the same word/phrase to express different meanings; and the classical Arabic is the language of the ancient people, the Holy Quran and Arabic classical books were written using this language.

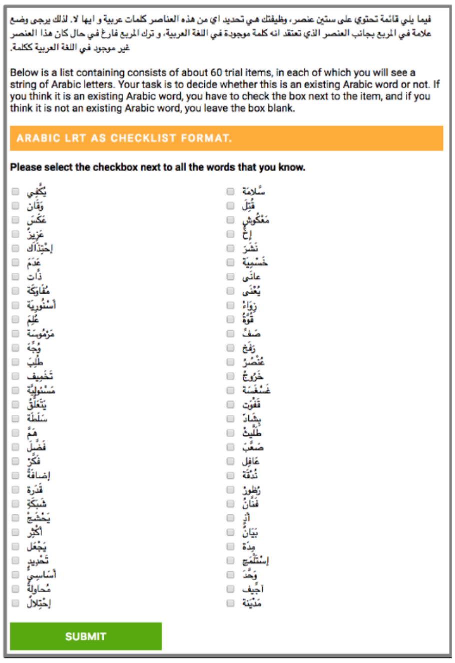
Some research contributions to Arabic natural language processing argued that Arabic lacks efficient approaches to measure Arabic learning proficiency using simple, fast, and efficient placement tests. Arabic Lexical Recognition Tests (ALRT), modules, applications and tools are still under development stages, (Salah et al., 2022); (Hamed, 2019), (Hamed and Zesch, 2018). According to (Hamed and Zesch, 2017), the Lexical Recognition Test (LRT) is a vocabulary size test, which is frequently used to calculate the number of words known by or acquired by a language learner. In such a test, the language learner is shown a list of vocabularies, and for each vocabulary, he/she needs to determine whether it is a valid word or nonword, and the LRT scores can be easily measured based on the learner's responses. Figure 1 shows a sample item of this test for both English and German. The main advantages of the LRT are it is simple, fast, and efficient. A test examiner roughly needs several minutes to answer all questions. LRTs come in two formats: a set of Yes/No questions or a customized checklist format.

Like English and German, as the number of Arabic learners increase, the necessity to have such kind of Arabic placement tests (LRT) increases as well. Currently, Arabic learning centers lack such kind of effective approaches to measure learners proficiency level. Thus, this research is a further step of our more recent work aiming at developing an Arabic LRTs tool, called ALRT (Arabic Lexical Recognition Tool. In the work (Hamed, 2019; Salah et al., 2022), we proposed a generic framework for the automatic generation of Arabic LRT, and developed an algorithm that follows some rules to generate high-quality nonwords that can confuse language learners and add certain levels of complexity, thus they are good distractors. Furthermore, this method applies some paradigms based on (i) statistical machine learning such as character n-gram models, and (ii) Arabic language special



(a) English (b) German

Figure 1: Examples of Yes/No questions



فيما يلي قائمة تحتوي على ستين عنصر، وظيفتك هي تحديد اي من هذه العناصر كلمات عربية و ايها لا. لذلك يرجى وضع علامة في المربع بجانب العنصر الذي تعتقد انه كلمة موجودة في اللغة العربية، و ترك المربع فارغ في حال كان هذا العنصر غير موجود في اللغة العربية ككلمة.

Below is a list containing consists of about 60 trial items, in each of which you will see a string of Arabic letters. Your task is to decide whether this is an existing Arabic word or not. If you think it is an existing Arabic word, you have to check the box next to the item, and if you think it is not an existing Arabic word, you leave the box blank.

ARABIC LRT AS CHECKLIST FORMAT.

Please select the checkbox next to all the words that you know.

□ يُكْفِي
□ وَقَان
□ عَكَسَ
□ عَزِيزٌ
□ إِحْتِذَاك
□ عَدَمَ
□ ذَات
□ مُفَاوَكَة
□ أَسْنُورِيَة
□ عُلِمَ
□ مَرْمُوسَة
□ وُجّهَ
□ طَلِبَ
□ تَخَمِيف
□ مَسْئولِيَّة
□ يَتَعَلَّقُ
□ سَلَطَة
□ هَمَّ
□ فَضَّلَ
□ فَكَّرَ
□ إِضَافَةٌ
□ قُدْرَة
□ شَبكَة
□ يَخْشَجْ
□ أَكْثَر
□ يَجْعَل
□ تَحْرِيد
□ أَسَاسِي
□ مُحَاوَلَة
□ إِخْتِلَالُ
□ سُلَامَة
□ قُتِلَ
□ مَعْكُوش
□ إِخْ
□ نَشَرَ
□ خَسْمِيَة
□ عانَى
□ يُعْنَى
□ زَوَاءُ
□ قُوَّةٌ
□ صَفٌ
□ رَفَخ
□ عُنْصُرٌ
□ خَرُوجُ
□ غَسْغَسَة
□ قَفْوَت
□ بِشَادَ
□ طُلِّيثْ
□ صَعُّبَ
□ عَاقِل
□ نُدْقَة
□ رُظُورْ
□ قَنَانٌ
□ أَثِرٍ
□ بَيَانٌ
□ مِدَة
□ إِسْتَلْمَج
□ وَحَّدَ
□ آجِّيف
□ مَدْيَنة

Figure 2: An example of Arabic LRT (ALRT)

characteristics such as orthography and phonological similarity maps. Finally, we applied some additional language features using word frequency map to generate multiple levels of Arabic LRT. Its worth noting that the adopted approach is mainly based on similar approaches that were applied on some European languages such as Spanish and German. Figure 2 shows a sample output of ALRT.

The remaining parts of the paper are structured as follows. The most relevant contributions are discussed in Section 2. Section 3 discusses some potential applications of using the ALRT tool. The current state of the tool is presented in Section 4. Finally, Section 5 concludes the paper and provides some ongoing research lines.

2 Related Work

Recently, measuring language proficiency levels has been attractive for many researchers. The lexical project (Balota et al., 2007) is one of the main contributions in this field, it is the common criterion being widely used to measure learning proficiency levels, it contains many international standard tests for any specific language. For example, the International English Language Testing System (ILETS), and the Test of English as a Foreign Language (TOEFL). Both tests are adopted by English-native countries to measure English language proficiency levels for official use such as business, work, academic, and international mobility, among others. The Lexical Recognition test (LRT) is another example, which is a short and quick test that is frequently used to estimate learners' proficiency for some international languages such as English, German, Spanish, and other Latin languages. Many related experiments, research, and contributions coming from various European centers have approved this concept with the help of real test beds and datasets. In the following, we shed light to the most relevant contributions for Arabic and discuss their main drawbacks. Consequently, we avoid the potential issues related to similar experiments that were conducted to design this form of tests previously. Also, we avoid some literature review associated with Arabic diacritics during the process of generating good nonwords like (Hamed and Zesch, 2017).

LexTALE is another criterion used to test language proficiency for English and German languages (Lemhöfer and Broersma, 2012). LexTALE is a five-minute, YES and NO vocabulary identification test. In its default settings, it consists of 60 questions, two-thirds are words and one-third are nonwords. Its performance shows good results when applied on a processed dataset of vocabularies. However, compared to other tests like the Test of English for International Communication (TOEIC), it is still substantial.

In Arabic, nonwords were manually generated by language experts who follow certain rules to generate high quality nonwords. This process is inefficient, time consuming and sometimes subjects to human errors. As the quality of nonwords plays a crucial role in determining accurate scores, high quality nonwords must be very similar both phonologically and orthographically to real words to increase the complexity of identifying them easily. LexTALE is a valid test that was adapted by other languages like German, French, and Spanish, and it can be used as a good measurement criterion for non-native language speakers who have various learning levels - small, medium, and high. (Duyck et al., 2004).

Generating the nonwords manually was also applied by English Lexicon Project (ELP) (Balota et al., 2007). It is a huge repository of language resources and databases both descriptive and behavioral, connected with a search engine that supplies the researchers with all resources they need to tackle any technical issue and obstacle they face during the process of implementing the lexical tests. Technically, the ELP is totally built using manual procedures to generate nonwords. This process is done by applying certain roles and language characteristics to replace one or more characters in a word with others to create a nonword with high similarity index to the original one considering orthographic and phonological characteristics. A similar work was applied to the British-English language (Rastle et al., 2002). The ARC nonword database was used by applying a generation model based on both phonological and orthographic rules. This ARC database was used to design the LRT test that tricks the learner in multiple ways based on the morphological, orthographic, and phonological rules.

The Wuggy research project (Keuleers and Brysbaert, 2010) developed a computer-based application that facilitates the process of generating nonwords automatically, it creates high quality pseudo words or nonwords following certain rules of languages, features, sub-syllabic structure, and transition frequencies among sub-syllabic elements. It is available for many languages such as English, Spanish, French, German, Basque, and Serbian. It could be applied to other languages with some extra efforts. In this regard, a pseudo word is given a more attention and can be taken as another important factor for determining the efficiency of the lexical decision, which represents a good tool by psycholinguists who perform word processing tasks. The Wuggy algorithm has some limitations (i) its dependency on sub-syllabic or summed bi-gram similarities decreases its performance; (ii) it is not a fully automated solution for nonword generation, it requires some human intervention to write the matching expressions; (iii) the algorithm has some technical issues in auto detecting the end of the given expression.

WordGen is another application which is similar to Wuggy, it is an automated tool used to generate and select nonwords for English, French, and German (Duyck et al., 2004). Here, both automatic and manual methods have been collaboratively used to generate nonwords. Other researchers (Hamed and Zesch, 2015), (Hegazi, 2016) argued on the importance of the role of Arabic diacritized in vocabulary assessments in the LRT, as they claimed that diacritization adds a new level of complexity and reveals ambiguity that introduces better evaluation for learners in identifying the words. Consequently, a sample test using both the diacritized version of Arabic LRT and the non-diacritized version was generated to show the importance of Arabic diacritization compared to other languages. The results showed that the absence of Arabic diacritization increases the ambiguity of word recognition. It is worth noting that the majority of Arabic written text is non diacritized, except in some religious, historical, classical books, and in some specialized Arabic educational fields. Diacritization impacts the design of nonwords as Arabic diacritization is an orthographic way to describe Arabic word pronunciation (Hamed and Zesch, 2017). They assumed that the non-diacritized nonwords are highly probably more difficult to guess than the diacritized ones. The diacritized nonwords can easily distract the language's learners when having more closely related words, especially if they come with labels having pronounceable diacritics.

In (Hamed and Zesch, 2015), Hamed and Zesch suggested the use of a fully automated methodology to generate high-quality nonwords for English LRTs. To implement the automated process of generating nonwords in English, they conducted some experiments to generate good nonwords using some methods based like Markov and character language models that automatically replace a letter with similar one. They also applied some mechanisms to rank the generated nonwords and used the highest ones in creating English LRT.

Similarly, in (Rastle et al., 2002), the authors developed an automatic paragdimg to generate nonwords for English Language. They constructed a database of nonwords based on both phonetic and orthographic language properties.

3 Applications of ALRT

The authors in (Gueddah and Yousfi, 2013) proposed an approach to improve Arabic spell checking in typing text. They suggested the use of a statistical model based on a similarity matrix to find Arabic letters' similarity degrees, this way each

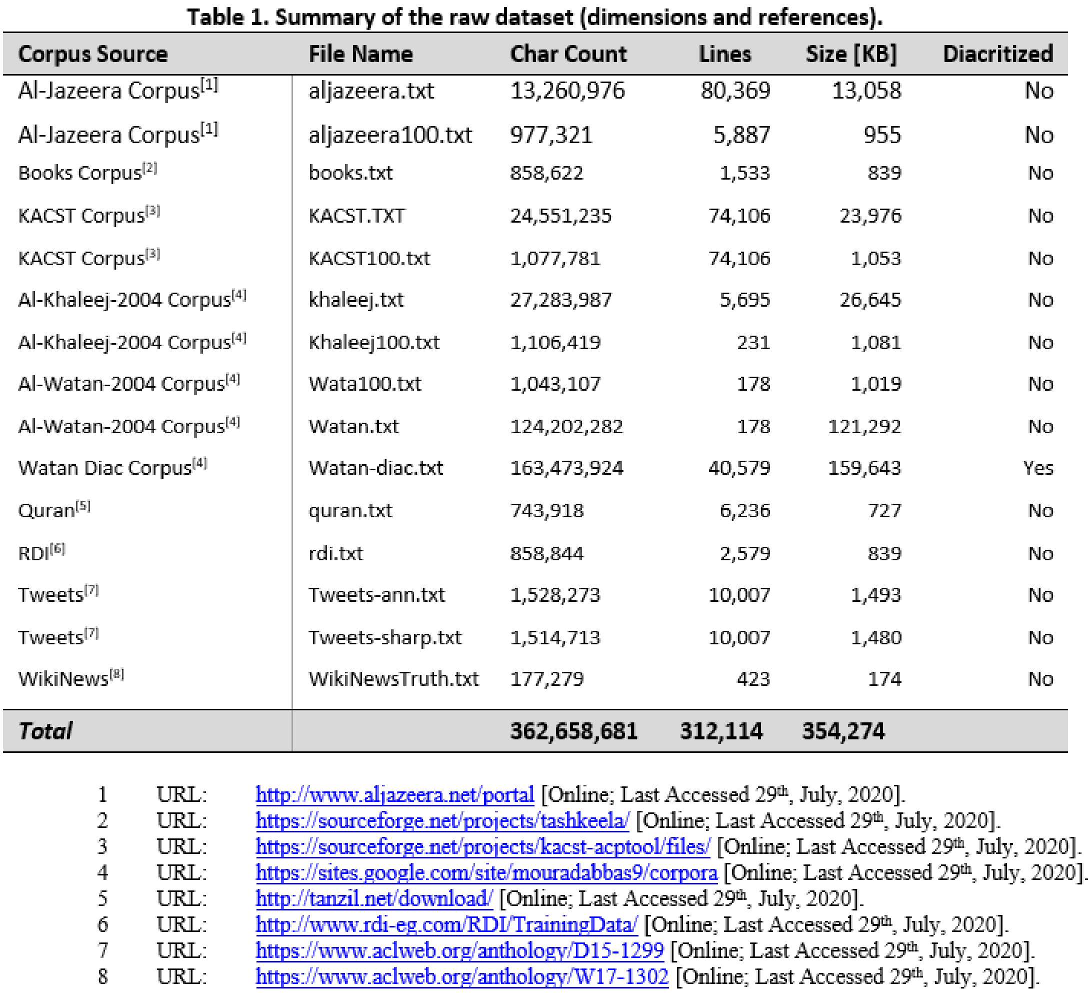


Table 1. Summary of the raw dataset (dimensions and references).

Corpus Source	File Name	Char Count	Lines	Size [KB]	Diacritized
Al-Jazeera Corpus[1]	aljazeera.txt	13,260,976	80,369	13,058	No
Al-Jazeera Corpus[1]	aljazeera100.txt	977,321	5,887	955	No
Books Corpus[2]	books.txt	858,622	1,533	839	No
KACST Corpus[3]	KACST.TXT	24,551,235	74,106	23,976	No
KACST Corpus[3]	KACST100.txt	1,077,781	74,106	1,053	No
Al-Khaleej-2004 Corpus[4]	khaleej.txt	27,283,987	5,695	26,645	No
Al-Khaleej-2004 Corpus[4]	Khaleej100.txt	1,106,419	231	1,081	No
Al-Watan-2004 Corpus[4]	Wata100.txt	1,043,107	178	1,019	No
Al-Watan-2004 Corpus[4]	Watan.txt	124,202,282	178	121,292	No
Watan Diac Corpus[4]	Watan-diac.txt	163,473,924	40,579	159,643	Yes
Quran[5]	quran.txt	743,918	6,236	727	No
RDI[6]	rdi.txt	858,844	2,579	839	No
Tweets[7]	Tweets-ann.txt	1,528,273	10,007	1,493	No
Tweets[7]	Tweets-sharp.txt	1,514,713	10,007	1,480	No
WikiNews[8]	WikiNewsTruth.txt	177,279	423	174	No
Total		**362,658,681**	**312,114**	**354,274**	

1 URL: http://www.aljazeera.net/portal [Online; Last Accessed 29th, July, 2020].
2 URL: https://sourceforge.net/projects/tashkeela/ [Online; Last Accessed 29th, July, 2020].
3 URL: https://sourceforge.net/projects/kacst-acptool/files/ [Online; Last Accessed 29th, July, 2020].
4 URL: https://sites.google.com/site/mouradabbas9/corpora [Online; Last Accessed 29th, July, 2020].
5 URL: http://tanzil.net/download/ [Online; Last Accessed 29th, July, 2020].
6 URL: http://www.rdi-eg.com/RDI/TrainingData/ [Online; Last Accessed 29th, July, 2020].
7 URL: https://www.aclweb.org/anthology/D15-1299 [Online; Last Accessed 29th, July, 2020].
8 URL: https://www.aclweb.org/anthology/W17-1302 [Online; Last Accessed 29th, July, 2020].

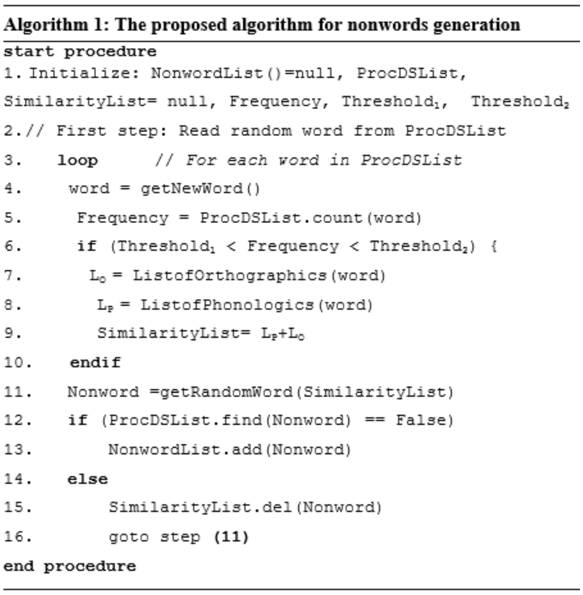
Arabic letter has a matrix of weighted degrees of similarities with other Arabic letters by assigning costs to the permutation errors generated by using the proximity degrees of keyboard characters and the calligraphic similarity in Arabic alphabet. Their aim was to develop a spell checking tool for malformed words that are created during the writing of Arabic documents. In a comparison to this work, we found another similarity matrix for each Arabic letter based on Arabic orthographic and phonological characteristics, so reputations will be performed based on a small set of similarities. Although the two works have different scopes, the main objective is to have an Arabic LRT that can be used as Arabic spellchecker. Compared to previous research contributions, this research work develops ALRT tool which is based on a proposed approach that considers generating the nonwords in a fully automated process using a newly developed algorithmic that implements some Arabic language characteristics such as spelling, orthography, pronunciation, phonology, n-grams, and the word frequency map which is mainly used to create multiple complexity levels of LRT test. In this regard, it wroth noting that to generate nonwords, we have been inspired by their definition: “words that fulfill the phonological constraints of the language but do not bear the meaning” (Huibregtse et al., 2002).

Another approach to generate nonwords in English is using minimal pairs (Ricks, 2015), a corresponding way to implement this concept in Arabic is the use of orthography and phonology roles. (Hamed and Zesch, 2015) argued that frequent n-grams are highly likely to generate high quality nonwords, which look like real words, and words that appear more frequently are easier to remember than less frequent words (Ellis, 2002). In addition to that some generated nonwords in Arabic could be classified as fake Arabic vocabulary that look like real words that were designed to distract the learners and confuse them in terms of phonetic if they tried a pronunciation or an orthographic letter that differ in terms of word writing shape.

In summary, to get a better picture of the practical value of the developed tool, we shed the light on three potential applications: First, since LRT themes are common methods to measure language learners' proficiency levels. However, the existing LRTs research for Arabic still has room for improvement, with few existing proposals at development stages, or existing proposals that mainly use human-crafted methods, or semi-automated methods using Arabic NLP techniques. Thus, an interesting application of the developed tool is to measure the proficiency level of Arabic learners (Arabic LRT). Fig. 3 shows an example of Arabic LRT that fits on one page. Second, another potential application is the Arabic spellchecker. Since the proposed approach can potentially generate a huge amount of good nonwords, these nonwords can be incorporated into any Arabic Proofreading tool that can be used as a reference model for spell-checking documents written in the Arabic for checking consistency, accuracy, and readability to meet professional standards. Third, since Arabic LRTs are still in the development stages, the proposed approach can be used as a reference by Arabic language researchers, who want to conduct relevant studies. The source code, the implementation steps, the documentation, and the generated nonwords database will be freely available on the GitHub platform. For now, we have uploaded the LRT test engine (https://github.com/ohamed/ar-lrts).

4 Current State of the Tool

The current version of the ALRT is V1.0, it is the initial draft that was built based on our previous work ((Salah et al., 2022)). Recently, we have proposed a generic framework for the automatic generation of Arabic LRT, and developed an algorithm that follows certain rules, and features to generate high quality nonwords with high similarity index to the original ones, and introduces certain levels of complexity to the LRT. In this work, we used a freely available corpora datasets that were collected from different resources, such as Arabic books, social media, and news agencies. It has a huge volume of Arabic texts in raw format that were transformed to one UTF-8 format having one vocabulary per line. Some preprocessing steps were also applied to make the data format suitable to work with. In data preprocessing, we mainly applied some data cleaning operations to remove special symbols, non Arabic characters, punctuation marks, numeric values, white-spaces, and any other strange character. Table 1 lists some technical features about the dataset. Column (1) represents the main corpus source; the available source of the data, some sources might have multiple files (rows in the table), number of alphabets, lines as in a notepad++ text file, size in Kilobytes (KB), whether the text is diacritized or not diacritized, and the main reference. Figure 3 shows the proposed block diagram of Arabic Lexical Recognition Test (LRT) ((Salah et al., 2022)), which is the tool we developed to generate Arabic nonwords.



Algorithm 1: The proposed algorithm for nonwords generation

```
start procedure
1. Initialize: NonwordList()=null, ProcDSList,
SimilarityList= null, Frequency, Threshold$_1$,  Threshold$_2$
2.// First step: Read random word from ProcDSList
3.   loop     // For each word in ProcDSList
4.    word = getNewWord()
5.     Frequency = ProcDSList.count(word)
6.     if (Threshold$_1$ < Frequency < Threshold$_2$) {
7.       L$_O$ = ListofOrthographics(word)
8.        L$_P$ = ListofPhonologics(word)
9.        SimilarityList= L$_P$+L$_O$
10.    endif
11.    Nonword =getRandomWord(SimilarityList)
12.    if (ProcDSList.find(Nonword) == False)
13.        NonwordList.add(Nonword)
14.    else
15.        SimilarityList.del(Nonword)
16.        goto step (11)
end procedure
```

4.1 Nonwords generation - Orthographic and phonological

The process of automatic generation of Arabic nonwords is based on the common Arabic language features, such as orthographic, phonological, n-grams, and vocabulary frequency. Algorithm 1 describes the pseudo-code for generating the nonwords. The proposed algorithm beings by iterating through all processed vocabularies found in the database. For each vocabulary, the algorithm calculates its frequency. To generate multilevel LRTs, the algorithm computes the word's frequency (how many times the chosen word appeared in the corpus). To tune the algorithm's operation in terms of words' frequencies, we used two thresholds - Threshold1 and Threshold2. If Frequency > Threshold1 && Frequency < Threshold2, we assume that the given vocabulary is not used more frequently. Two lists will be created, one contains the orthographic vocabularies using orthographic similarity roles, and the other contains phonological vocabularies using phonological similarity map. next, the two list are merged to construct the similarity list that includes all vocabularies. The algorithm randomly selects a set of vocabularies from the similarity list "SimilarityList" and checks the occurrence of them in the processed data. If the selected vocabulary is a real word, it will be removed from the similarity list. The algorithm repeats the process to select a new vocabulary.

4.2 N-grams generation

To further improve the automatic generation of nonwords, the results of Algorithm 1 have been updated by implementing the character n-grams concepts that represent the subsequent characters of vocabulary. This process iterates through the processed data file, and then for each vocabulary, it generates all possible n-grams starting from bigram to word-length-1 grams. These n-grams were appended to the database table along side with their corresponding real words, this step is useful in formulating a statistical data reference for which conclusions and judgements can be built easily. Since n-grams could be involved in generating nonwords by replacing a character in the input word taking into consideration frequency occurrence of prefix and postfix characters. Consequently, the closet character from the similarity set intersected with a character that uses frequency in the n-grams list will be substituted. This way, n-grams are being used to narrow the acceptable possibilities; this is expected to improve the quality of the nonwords generation process.

5 Conclusion and Future Work

In this paper, we have introduced the Arabic Lexical Recognition Tests Tool (ALRT) for the automatic generation of Arabic LRTs. The proposed tool will automatically generate nonwords based on a newly proposed model, which considers Arabic special characteristics such as orthography (spelling), phonology (pronunciation), n-grams, and the word frequency map, which is an important factor to create a multi-level test. The tool was tested using a huge dataset of Arabic vocabulary, and a human-driven intervention was used as an extra verification step to validate the quality of generated nonwords. We are working on integrating the ALP (Freihat et al., 2018b,a) lemmatizer for generating lemmas automatically. We also plan to add other Tests to the tool such as tokenization recognition tests, part of speech recognition, and diacritization recognition tests.

References

Ehsan Mohammed Abdelgadir and VSV Laxmi Ramana. 2017. *A Handbook on "Introduction to Phonetics & Phonology": For Arabic students*. Notion Press.

David A Balota, 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.

Wouter Duyck, 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.

Heba Elfardy and Mona Diab. 2012. Aida: Automatic identification and glossing of dialectal arabic. In *Proceedings of the 16th eamt conference (project papers)*, pages 83–83.

Nick C Ellis. 2002. Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in second language acquisition*, 24(2):143–188.

A. A. Freihat, M. Abbas, G. Bella, and F. Giunchiglia. 2018a. Towards an optimal solution to lemmatization in arabic. In *Proceedins of the 4th International Conference on Arabic Computational Linguistics (ACLing 2018)*, pages 1–9.

A. A. Freihat, G. Bella, H. Mubarak, and F. Giunchiglia. 2018b. A single-model approach for arabic segmentation, pos tagging, and named entity recognition. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–8.

Hicham Gueddah and Abdallah Yousfi. 2013. The impact of arabic inter-character proximity and similarity on spell-checking. In *2013 8th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–4. IEEE.

Osama Hamed and Torsten Zesch. 2015. Generating nonwords for vocabulary proficiency testing. In *Proceeding of the 7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 473–477.

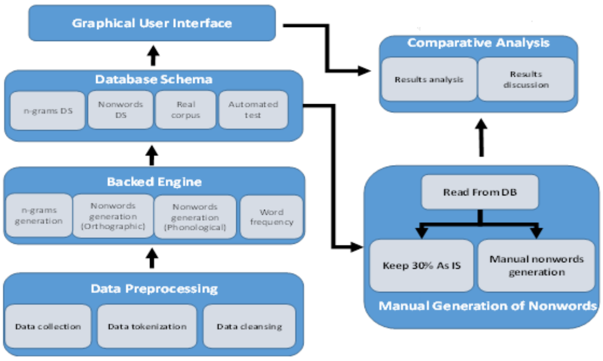


Figure 3: The proposed block diagram of Arabic Lexical Recognition Test (LRT)

Osama Hamed and Torsten Zesch. 2017. The role of diacritics in designing lexical recognition tests for arabic. *Procedia Computer Science*, 117:119–128.

Osama Hamed and Torsten Zesch. 2018. The role of diacritics in adapting the difficulty of arabic lexical recognition tests. *NLP for Computer Assisted Language Learning (NLP4CALL 2018)*, 23.

Osama Amin Hamed. 2019. *Automatic generation of lexical recognition tests using natural language processing*. Ph.D. thesis, Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2019.

Mohamed Osman Hegazi. 2016. An approach for arabic root generating and lexicon development. *Int. J. Comp. Sci. Netw. Sec.(IJCSNS)*, 16(1):9.

Ineke Huibregtse, Wilfried Admiraal, and Paul Meara. 2002. Scores on a yes-no vocabulary test: Correction for guessing and response style. *Language testing*, 19(3):227–245.

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

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

Kathleen Rastle, 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.

Robert Stephen Ricks. 2015. *The development of frequency-based assessments of vocabulary breadth and depth for L2 Arabic*. Georgetown University.

Saeed Salah, Mohammad Nassar, Raid Zaghal, and Osama Hamed. 2022. Towards the automatic generation of arabic lexical recognition tests using orthographic and phonological similarity maps. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8429–8439.